COMPUTER SIMULATION OF EQUILIBRIUM MULTICOMPONENT MULTISTAGE SEPARATION PROCESSES

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

SAHIDUL ISLAM

to the

DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MAY, 1985

31.5.85

CERTIFICATE

This is to certify that the work presented in this thesis entitled," COMPUTER SIMULATION OF EQUILIBRIUM MULTICOMPONENT MULTISTAGE SEPARATION PROCESSES " has been carried out by Mr. Sahidul Islam under my supervision and the same has not been submitted elsewhere for a degree.

Prahlada Rao.
D.P. Rao
Assistant Professor
Department of Chemical Engg.
Indian Institute of Technology
Kanpur - 208016

May 1985

ACKNOWLEDGEMENT

I feel pleasure to express my deep appreciation to Dr. D.P. Rao for his invaluable help for the completion of this thesis.

I would be extremely happy to convey my sincere thanks to all my friends and seniors in different fields who helped me directly and indirectly for the completion of this thesis. Special thanks are due to Dr. Ratan Mohan, Dr. A.K. Verma, Mr. Raghu Raman, Mr. Mrityunjoy Chakroborty, Mr. Binoy Bhusan Kandir, Dr. Sujit Dutta, Mr. Susil Mandal, Mr. Kausik Banerjee, Mr. Tapas Mandal, Mr. S.R. Diksitulu, Mr. Alok Pandit, Mr. VLN Murthy, Mr. Saibal Banerjee, Mr. M.M. Beg, Dr. P.K. Bhatt and Miss Arti Gupta.

I do not know how I will express my deep appreciation to our immortal beloved Newton. As a child I am trying to understand and to apply one of his principle, which is nothing compare to his major contributions, but is the heart of this thesis.

Finally what I feel that help and inspiration is as if solely from * Marrya who boosted me and thereby helping me to think the problem, has been inducing and driving me to my ultimate long desire.

Sahidul Islam

CONTENTS

ABSTRACT

Name	:	Sahidul Islam
Programme	:	M.Tech.
Department	:	Chemical Engineering
Institute	:	Indian Institute of Technology Kanpur - 208 016, INDIA
Month & Year of submission	:	May 1985
Title of the Thesis	:	Computer Simulation of Equilibrium Multicomponent Multistage Separation Processes
Thesis Supervisor	:	Dr. D.P. Rao

The modified Thomas algorithm is employed to obtain the correction vector ΔX in the widely used Naphthali-Sandholm method of solving separation processes problems. An efficient algorithm for obtaining the correction ΔX is proposed in which the sparsity of the submatrices of the Jacobian has been expoited in the matrix multiplication and inversion. The operation count for the proposed algorithm and the standard matrix multiplication and inversion has been presented using two bench-mark problems. It has been shown that the use of the proposed algorithm results in considerable saving in the CPU time with increase in the number of components and stages. The proposed algorithm can be used for the extensions and variants of Naphthali-Sandholm methods.

CHAPTER 1

INTRODUCTION

The classical techniques of separation of multisomponent mixtures like distillation, absorption, extraction, etc. are widely used in chemical industries. The design or simulation of these (stage-wise) separation processes involves the solution of the material and energy balance equations and equilibrium relations for each stage, and requires an enormous amount of computational effort. Before the widespread use of computers, short-out methods were employed for the design though these are generally inadequate for systems other than the ones for which the equilibrium relations are linear. But with accessability to powerful computers, the rigorous methods got impetus and several methods have been proposed.

The rigorous methods of design or simulation involve two major steps; namely formulation of basic equations and their numerical methods of solution. The methods of formulation of the basic equations, in turn, can be classified as componentwise grouping of variables and stage-wise grouping of variables. The earlier methods of solution like the B-P method, the sum-rate method and the relaxation technique can be visualized as the direct substitution methods. Recently, these methods of solution have given way to the more efficient Newton-Raphson method and its variants.

Under.the component-wise grouping of variables,the basic equations can be obtained as

$$\overline{\overline{C}}_i \; X_i = - \overline{F}_i \qquad \text{for} \quad i = 1,\ 2 \ldots\ldots C \tag{1}$$

for the steady state conditions; where $\overline{\overline{C}}_i$ is the coefficient matrix involving vapor and liquid flow rates and the k-values, $\overline{X}_i$ could be either liquid or vapor mole fractions or component flow rates, $\overline{F}_i$ are the feed component flow rates and i the components. In the relaxation technique (also known as False-Transient method) the basic equations are cast as

$$\overline{\overline{C}}_i^{\,k+1} \; \overline{X}^{\,k+1} = \overline{X}_i^{\,k} \qquad i = 1,2\ldots.C \tag{2}$$

where k is the iteration number.
More generally, the basic equations may be represented as

$$\overline{\overline{A}}_i \; \overline{X}_i = \overline{b}_i \qquad i = 1,2\ldots\ldots C \tag{3}$$

The square matrix has the tridiagonal structure and $\overline{X}_i$ can be found using the well known Thomas algorithm.

Equation (3) together with the enthalpy balance equations around the stages can be solved by the direct substitution method for any assumed liquid and vapor flow rates and temperatures. The B-P method, the sum-rate method and the modified relaxation technique fall under this category.

Since the year 1965, the direct substitution methods gave way to the Newton-Raphson method(or its modified versions) of solution. In these methods, the material balance discrepancy functions,

$$M_n = \sum_{i=1}^{C} y_{i,n} - \sum_{i=1}^{C} x_{i,n} \qquad n = 1,2\ldots .N \tag{4}$$

and the enthalpy discrepancy functions around each stage are expanded in the Taylor series. Assuming the second and higher order terms to be negligible, the equations are rearranged to obtain

$$\bar{\bar{J}} \; \Delta \bar{X} = \bar{b} \tag{5}$$

where $\bar{\bar{J}}$ is the Jacobian matrix of 2Nx2N.
Several convergence schemes and the methods of obtaining the correction vector have been proposed. These have been discussed by Holland (2) .

Naphthali (4) and later Naphthali and Sandholm (5) showed that the convergence characteristics are better if the stage-wise grouping of variables together with the Newton-Raphson method are employed. For each of the stages the variables are the component liquid flow rates, temperature and the component vapor flow rates (i.e. l_i, T, v_i). The component and the enthalpy discrepency functions are formulated to yield

$$\bar{F}\,(\bar{X}) = 0$$

Applying the Newton-Raphson technique we get

$$\bar{\bar{J}}\,\overline{\Delta X} = -\bar{F}$$

The Jacobian is of the order (2C+1) N x(2C+1)N and it has the block tridiagonal form. Several methods proposed for solving the separation problems deal with the techniques of obtaining the correction vector , Stadther[6, 7, 8} .

It has been generally accepted that the Naphthali-Sandholm method has better convergence characteristics. However, it is likely to diverge if the guessed component liquid and vapor flow rates and temperatures are far from the correct values. To over come this problem, Ketchum [3] proposed the fusion of the relaxation technique and the Naphthali-Sandholm method and demonstrated its suitablity for even interlinked columns. Later Hofeling and Seader [1] have demonstrated as to how the modified Thomas algorithm can be used for the interlinked columns. Stadther [6,7,8] has compared the convergence characteristics of the modified Thomas algorithm and the other methods of solution using the sparse matrix technique. He finds the Thomas algorithm as good as the other sparse matrix techniques.

From the literature, it appears that the sparsity of the submatrices (diagonal and its adjacent submatrix) has not been exploited. The objective of the present work is to propose a more efficient method of solution taking advantage of the sparsity of the submatrices.

In Chapter 2, the method of formulation of the basic equations and the method of solution is presented. The results and discussion are presented in Chapter 3. Next conclusions is presented.

CHAPTER 2

BASIC EQUATIONS AND METHOD OF SOLUTION

In this Chapter, the strategy of formulating the basic equations involving material and enthalpy balances, equilibrium and tray efficiency relations of multistage multicomponent separation processes, and the method of obtaining solution is presented.

A model of a general stage, along with the notation employed herein, is depicted in figure 1. The various discrepancy functions can be formulated as follows:

For Material balance:

$$M_{i,n} = l_{i,n-1} + v_{i,n+1} + f_{i,n} - (1+S_n)v_{i,n} - (1+s_n)l_{i,n} \qquad (1)$$

for $1 \leq i \leq C$ & $1 \leq n \leq N$

where $v_{i,n}$ and $l_{i,n}$ are the flow rates of component i in liquid and vapor leaving the nth stage, and S_nV_n and s_nl_n are the vapor and liquid side streams drawn on the nth stage.

For Enthalpy balance:

$$E_n = \sum_{i=1}^{C} l_{i,n-1}h_{i,n-1} + \sum_{i=1}^{C} v_{i,n+1}H_{i,n+1} + \sum_{i=1}^{C} f_{i,n}h_{F_{i,n}} - \sum_{i=1}^{C}(1+S_n)v_{i,n}H_{i,n} - \sum_{i=1}^{C}(1+s_n)l_{i,n}h_{i,n} + q_n \ldots\ldots \qquad (2)$$

for $1 \leq i \leq C$ & $1 \leq n \leq N$

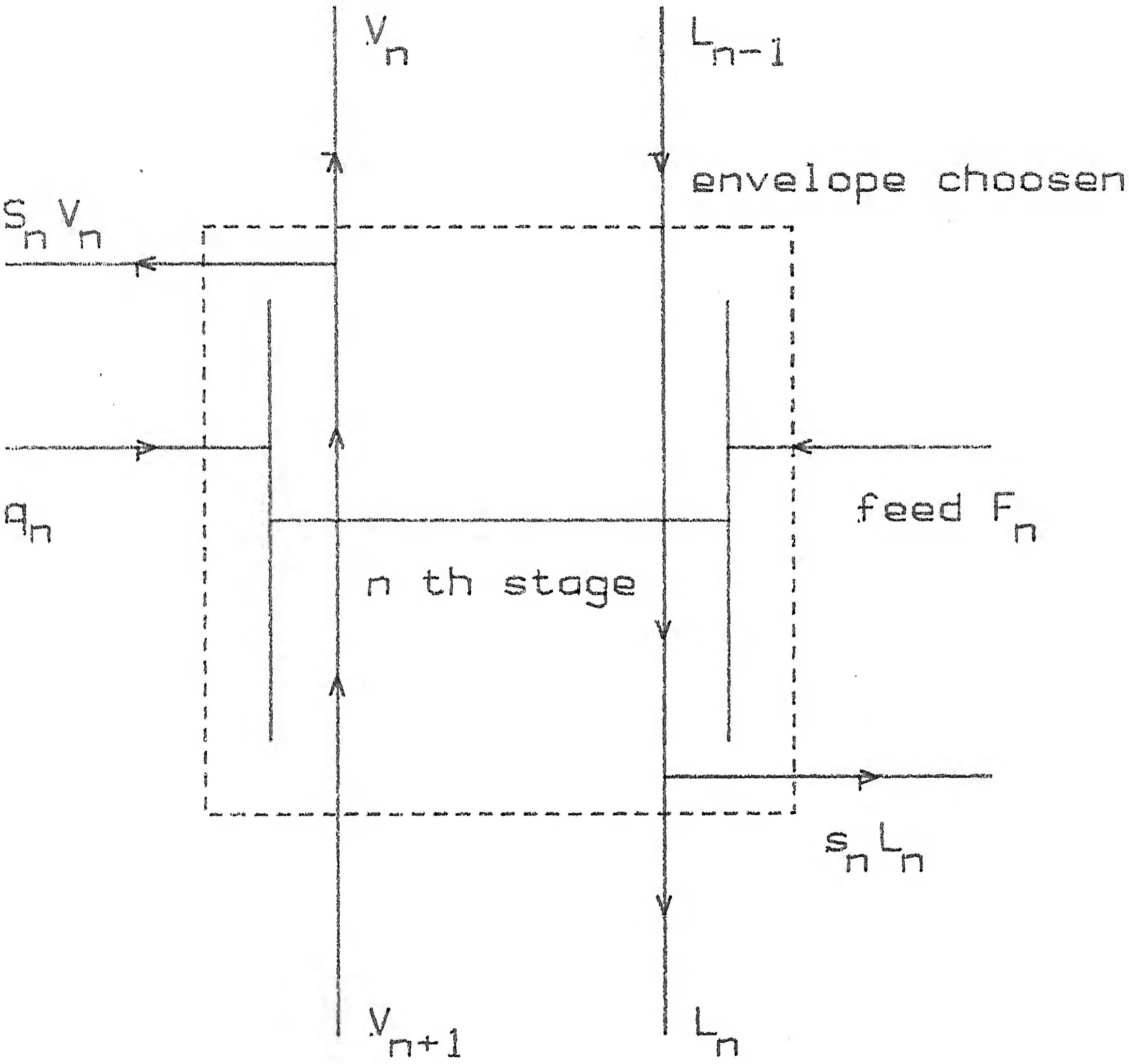

Fig. 1: A typical contacting stage

Equilibrium relations together with the Murphree efficiency:

$$O_{i,n} = \frac{\eta_n k_{i,n} l_{i,n}}{L_n} - \frac{v_{i,n}}{V_n} + (1-\eta_n) \frac{v_{i,n+1}}{V_{n+1}} \quad .... \qquad (3)$$

for $1 \leq i,j \leq C$ & $1 \leq n \leq N$

$$\text{where } \eta_n = \frac{y_{i,n} - y_{i,n+1}}{k_{i,n} x_{i,n} - y_{i,n+1}} \quad ..... \qquad (3a)$$

Thus, there are N(2C+1) set of nonlinear equations and they have, to be solved to obtain the N(2C+1) unknown variables, nam $l_{i,n}$, $v_{i,n}$ and T_n.

The set of equations may be compactly written as

$$\bar{F} = \bar{F}(\bar{X}) = \bar{0} \qquad (4)$$

where

$$\bar{F} = \left[\bar{F}_1, \bar{F}_2 .... \bar{F}_n .... \bar{F}_N\right]^T, \qquad (5)$$

$$F_n = \left[M_{1,n}, M_{2,n} \;....\; M_{C,n}, O_{1,n}, O_{2,n} .... O_{C,N}, E_n\right]^T \qquad (6)$$

$$\bar{X} = \left[\bar{X}_1, \bar{X}_2 .... \bar{X}_m\right]^T \qquad (7)$$

and $$\bar{X}_n = \left[l_{1,n}, l_{2,n} ...... l_{C,n}, v_{1,n}, v_{2,n} .... v_{C,n}, T_n\right]^T \qquad (8)$$

It may be pointed out that the ordering of the variables is slightly different from the one proposed by Naphthali and

Sandholm 1971 . The reasons for the order chosen here is explained later.

Employing the Newton-Raphson technique, from Equation(4) we can obtain

$$\Delta \bar{X} = - \left[\frac{\partial \bar{F}}{\partial \bar{X}} \right]^{-1} \cdot \bar{F}$$

$$= - \bar{\bar{J}}^{-1} \cdot \bar{F}$$

where $\bar{\bar{J}}$ is the Jacobian matrix.

The Jacobian has the block tridiagonal structure and can be represented as

$$\bar{\bar{J}} = \begin{bmatrix} \bar{\bar{B}}_1 & \bar{\bar{C}}_1 & & & & & \\ \bar{\bar{A}}_2 & \bar{\bar{B}}_2 & \bar{\bar{C}}_2 & & & & \\ & \bar{\bar{A}}_3 & \bar{\bar{B}}_3 & \bar{\bar{C}}_3 & & & \\ & & - & - & - & & \\ & & & \bar{\bar{A}}_n & \bar{\bar{B}}_n & \bar{\bar{C}}_n & \\ & & & & - & - & - \\ & & & & & \bar{\bar{B}}_N & \bar{\bar{C}}_N \end{bmatrix}$$

where

$$\bar{\bar{A}}_n = \frac{\partial \bar{F}_n}{\partial \bar{X}_{n-1}} , \quad \bar{\bar{B}}_n = \frac{\partial \bar{F}_n}{\partial \bar{X}_n} \quad \text{and} \quad \bar{\bar{C}}_n = \frac{\partial \bar{F}_n}{\partial \bar{X}_{n+1}}$$

and the rest of the elements are null matrices.

In expand form, the submatrix $\overset{=}{A}_n$ is

$$\overset{=}{A}_n = \left[\begin{array}{cccccccc}
\frac{\partial M_{1,n}}{\partial l_{1,n-1}}, & \frac{\partial M_{1,n}}{\partial l_{2,n-1}} & \cdots & \frac{\partial M_{1,n}}{\partial l_{C,n-1}} & \cdots & \frac{\partial M_{1,n}}{\partial v_{1,n-1}} & \cdot\cdot\frac{\partial M_{1,n}}{\partial v_{1,n-1}}, & \frac{\partial M_{1,n}}{\partial T_{n-1}} \\
\frac{\partial M_{2,n}}{\partial l_{1,n-1}} & \frac{\partial M_{2,n}}{\partial l_{2,n-1}} & \cdots & \frac{\partial M_{2,n}}{\partial l_{C,n-1}} & \frac{\partial M_{2,n}}{\partial v_{1,n-1}} & \cdots & \frac{\partial M_{2,n}}{\partial v_{C,n-1}} & \frac{\partial M_{2,n}}{\partial T_{n-1}} \\
\cdots & \cdots & & \cdots & \cdots & & & \cdots \\
\cdots & \cdots & & \cdots & \cdots & & & \cdots \\
\cdots & \cdots & & \cdots & \cdots & & & \cdots \\
\frac{\partial M_{C,n}}{\partial l_{1,n-1}} & \frac{\partial M_{C,n}}{\partial l_{2,n-1}} & \cdots & \frac{\partial M_{C,n}}{\partial l_{C,n-1}} & \frac{\partial M_{C,n}}{\partial v_{1,n-1}} & \cdots & \frac{\partial M_{C,n}}{\partial v_{C,n-1}} & \frac{\partial M_{C,n}}{\partial T_{n-1}} \\
\frac{\partial O_{1,n-1}}{\partial l_{1,n-1}} & \frac{\partial O_{1,n-1}}{\partial l_{2,n-1}} & \cdots & \frac{\partial O_{1,n-1}}{\partial l_{C,n-1}} & \frac{\partial O_{1,n-1}}{\partial v_{1,n-1}} & & \frac{\partial O_{1,n-1}}{\partial v_{C,n-1}} & \frac{\partial O_{1,n-1}}{\partial T_{n-1}} \\
\cdots & \cdots & & \cdots & \cdots & & & \cdots \\
\cdots & \cdots & & \cdots & \cdots & & & \cdots \\
\frac{\partial O_{C,n-1}}{\partial l_{1,n-1}} & \frac{\partial O_{C,n-1}}{\partial l_{2,n-1}} & & \frac{\partial O_{C,n-1}}{\partial l_{C,n-1}} & \frac{\partial O_{C,n-1}}{\partial v_{1,n-1}} & & \frac{\partial O_{C,n-1}}{\partial v_{C,n-1}} & \frac{\partial O_{C,n-1}}{\partial T_{n-1}} \\
\frac{\partial E_n}{\partial l_{1,n-1}} & \frac{\partial E_n}{\partial l_{2,n-1}} & & \frac{\partial E_n}{\partial l_{C,n-1}} & \frac{\partial E_n}{\partial v_{1,n-1}} & \cdots & \frac{\partial E_n}{\partial v_{C,n-1}} & \frac{\partial E_n}{\partial T_{n-1}}
\end{array}\right]$$

or, $$= \begin{bmatrix} \overset{=}{I}_C & \overset{=}{0}_C & \bar{0}_C \\ \overset{=}{0}_C & \overset{=}{0}_C & \bar{0}_C \\ \bar{h}_C & \bar{0}_C & C_{PL} \end{bmatrix}$$

where $\overset{=}{I}$, $\overset{=}{0}$, $\bar{0}$ are identity matrix, null matrix and null vector respectively. Subscript C indicates the dimension of the matrix or vector. $\bar{h}$ is the liquid enthalpy vector(h_i,i=1,C) and C_{PL} is heat capacity of L_{n-1}.

In expanded form, the submatrix $\overline{\overline{B}}_n$ is

$$\overline{\overline{B}}_m = \begin{bmatrix}
\frac{\partial M_{1,n}}{\partial l_{1,n}}, & \frac{\partial M_{1,n}}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial M_{1,n}}{\partial l_{C,n}}, & \frac{\partial M_{1,n}}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial M_{1,n}}{\partial v_{C,n}}, & \frac{\partial M_{1,n}}{\partial T_n} \\
\frac{\partial M_{2,n}}{\partial l_{1,n}}, & \frac{\partial M_{2,n}}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial M_{2,n}}{\partial l_{C,n}}, & \frac{\partial M_{2,n}}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial M_{2,n}}{\partial v_{C,n}}, & \frac{\partial M_{2,n}}{\partial T_n} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial M_{C,n}}{\partial l_{1,n}}, & \frac{\partial M_{C,n}}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial M_{C,n}}{\partial l_{C,n}}, & \frac{\partial M_{C,n}}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial M_{C,n}}{\partial v_{C,n}}, & \frac{\partial M_{C,n}}{\partial T_m} \\
\frac{\partial O_{1,n}}{\partial l_{1,n}}, & \frac{\partial O_{1,n}}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial O_{1,n}}{\partial l_{C,n}}, & \frac{\partial O_{1,n}}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial O_{1,n}}{\partial v_{C,n}}, & \frac{\partial O_{1,n}}{\partial T_n} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial O_{C,n}}{\partial l_{1,n}}, & \frac{\partial O_{C,n}}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial O_{C,n}}{\partial l_{C,n}}, & \frac{\partial O_{C,n}}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial O_{C,n}}{\partial v_{C,n}}, & \frac{\partial O_{C,n}}{\partial T_n} \\
\frac{\partial E_n}{\partial l_{1,n}}, & \frac{\partial E_n}{\partial l_{2,n}} & \cdots\cdots & \frac{\partial E_n}{\partial l_{C,n}}, & \frac{\partial E_n}{\partial v_{1,n}} & \cdots\cdots & \frac{\partial E_n}{\partial v_{C,n}}, & \frac{\partial E_m}{\partial T_n}
\end{bmatrix}$$

and can be simplified to yield

$$\bar{\bar{B}}_n = \left[\begin{array}{c:c} -(1+s_n)\ \text{only diagonal element of block submatrix } -(1+s_n) & -(1+S_n)\ \text{only diagonal elements of block submatrix same element } -(1+S_n) \\ \hdashline \text{Totally filled submatrix} & \text{Totally filled submatrix} \\ h_1 \ldots\ldots\ldots h_c \ \ldots & H_1 \ldots\ldots\ldots H_C \end{array}\right] *$$

$$\text{or, } \bar{\bar{B}}_n = \begin{bmatrix} -(1+s_n)\ \bar{\bar{I}}_C & (1+S_n)\ \bar{\bar{I}}_C & \bar{0}_C \\ \bar{\bar{X}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ (1+s_n)\ \bar{h}_C & (1+s_n)\bar{H}_C & \bar{C}_{PLV} \end{bmatrix}$$

where $\bar{\bar{X}}$, $\bar{X}$ and X denote the filled matrix, vector and nonzero element respectively and the subscript C the orders of the matrix or vector.

$$\overset{=}{C}_n = \left[\begin{array}{cccccccc}
\frac{\partial M_{1,n}}{\partial l_{1,n+1}}, & \frac{\partial M_{1,n}}{\partial l_{2,n+1}} & \cdots\cdots & \frac{\partial M_{1,n}}{\partial l_{C,n+1}}, & \frac{\partial M_{1,n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial M_{1,n}}{\partial v_{C,n+1}}, & \frac{\partial M_{1,n}}{\partial T_{n+1}} \\
\frac{\partial M_{2,n}}{\partial l_{1,n+1}}, & \frac{\partial M_{2,n}}{\partial l_{2,n+1}} & \cdots\cdots & \frac{\partial M_{2,n}}{\partial l_{C,n+1}}, & \frac{\partial M_{2,n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial M_{2,n}}{\partial v_{C,n+1}}, & \frac{\partial M_{2,n}}{\partial T_{n+1}} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial M_{C,n}}{\partial l_{1,n+1}}, & \frac{\partial M_{C,n}}{\partial l_{2,n+1}} & \cdots\cdots & \frac{\partial M_{C,n}}{\partial l_{C,n+1}}, & \frac{\partial M_{C,n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial M_{C,n}}{\partial v_{C,n+1}}, & \frac{\partial M_{C,n}}{\partial T_{n+1}} \\
\frac{\partial O_{1,n}}{\partial l_{1,n+1}}, & \frac{\partial O_{1,n}}{\partial l_{1,n+1}} & \cdots\cdots & \frac{\partial O_{1,n}}{\partial l_{1,n+1}}, & \frac{\partial O_{1,n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial O_{1,n}}{\partial v_{1,n+1}}, & \frac{\partial O_{1,n}}{\partial T_{n+1}} \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial O_{C,n}}{\partial l_{1,n+1}}, & \frac{\partial O_{C,n}}{\partial l_{2,n+1}} & \cdots\cdots & \frac{\partial O_{C,n}}{\partial l_{C,n+1}}, & \frac{\partial O_{C,n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial O_{C,n}}{\partial v_{C,n+1}}, & \frac{\partial O_{C,n}}{\partial T_{n+1}} \\
\frac{\partial E_{n}}{\partial l_{1,n+1}}, & \frac{\partial E_{n}}{\partial l_{2,n+1}} & \cdots\cdots & \frac{\partial E_{n}}{\partial l_{C,n+1}}, & \frac{\partial E_{n}}{\partial v_{1,n+1}} & \cdots & \frac{\partial E_{n}}{\partial v_{C,n+1}}, & \frac{\partial E_{n}}{\partial T_{n+1}}
\end{array} \right]$$

$$\bar{\bar{C}}_n = \left[\begin{array}{ccc} & \text{unit block submatrix} & \text{null column vector} \\ \text{null block matrix} & \text{Totally filled block sub matrix} & \\ & H_{1,n+1} \cdots\cdots\cdots\cdots H_{C,n+1} & * \end{array}\right]$$

and can be simplified to get

$$\left[\begin{array}{ccc} \bar{\bar{O}}_C & \bar{\bar{I}}_C & \bar{O}_C \\ \bar{\bar{O}}_C & \bar{\bar{X}}_C & \bar{O}_C \\ \bar{\bar{O}}_C & \bar{\bar{H}}_C & C_{AV} \end{array}\right]$$

THOMAS ALGORITHM

The correction vector $\Delta \bar{X}$ can be found by the Thomas algorithm and is given below

Forward Substitution:

For the stage 1

Steps: 1 $\bar{\bar{P}}_1 \longleftarrow (\bar{\bar{B}}_1)^{-1} \bar{\bar{C}}_1$

: 2 $\bar{Q}_1 \longleftarrow (\bar{\bar{B}}_1)^{-1} \bar{F}_1$

For the stages n from 2 to (N-1)

: 3 $\bar{\bar{P}}_n \longleftarrow (\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{P}}_{n-1})^{-1} \bar{\bar{C}}_n$

: 4 $\bar{Q}_n \longleftarrow (\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{P}}_{n-1})^{-1} (\bar{F}_n - \bar{\bar{A}}_n \bar{Q}_{n-1})$

For the last stage N

: 5 $\bar{Q}_N \longleftarrow (\bar{\bar{B}}_N - \bar{\bar{A}}_N \bar{\bar{P}}_{N-1})^{-1} (\bar{F}_N - \bar{\bar{A}}_N \bar{Q}_{N-1})$

Backward substitution

: 6 $\Delta \bar{X}_N \longleftarrow \bar{Q}_N$

For the stages n from N-1 to 1

: 7 $\Delta \bar{\bar{X}}_n \longleftarrow -(\bar{Q}_n - \bar{\bar{P}}_n \bar{X}_{n+1})$

It can be seen that the algorithm involves several time taking matrix multiplications and inversions. Advantage of the sparcity and the structure of these matrices can be taken to minimise the machine operations, as described below.

In step 1 and 3, we encounter the matrix multiplications $(\bar{\bar{B}})^{-1}\bar{\bar{C}}_1$ and $(\bar{\bar{B}}_j - \bar{\bar{A}}_j \bar{\bar{P}}_j)^{-1}\bar{\bar{C}}_j$. The inverted matrices are totally filled while $\bar{\bar{C}}_1$ and $\bar{\bar{C}}_j$ are highly sparse. The structure of the resulting matrix is

$$\underset{(\bar{\bar{B}})^{-1}\text{or }(\bar{\bar{B}}-\bar{\bar{A}}_j\bar{\bar{P}}_{j-1})}{\begin{bmatrix} \text{Filled} & \\ \text{matrix} & \\ & \end{bmatrix}} \underset{\bar{\bar{C}}_j}{\begin{bmatrix} \bar{\bar{O}}_C & \bar{\bar{I}}_C & \bar{O}_C \\ \bar{\bar{O}}_C & \bar{\bar{X}}_C & \bar{O}_C \\ \bar{O}_C & \bar{X}_C & X \end{bmatrix}} = \underset{\bar{\bar{P}}_j}{\begin{bmatrix} \bar{\bar{O}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{\bar{O}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{O}_C & \bar{X}_C & X \end{bmatrix}}$$

where $\bar{\bar{X}}$, $\bar{X}$ and X denote the filled matrix, vector and non-zero element respectively, $\bar{\bar{O}}$ and $\bar{\bar{I}}$ are the null and identity matrices, and the subcript C the order of the matrix or vector.

It can be seen that there is no need to compute the elements of the first C columns of the resulting matrix and we can save $C\,(2C+1)^2$ operations. Further, the next C columns can be found as

$$P_{j,C+i} = b^I_{j,i+C} + \sum_{K=C+1}^{2C+1} b^I_{j,k}\; C_{K,C+i} \quad \text{for} \quad \begin{matrix} 1 \leq j \leq 2C+1 \\ 1 \leq i \leq C \end{matrix} \qquad (3)$$

The elements of the last column are

$$P_{j,2C+1} = b^I_{j,2C+1}\; C_{2C+1,2C+1} \qquad (4)$$

Thus the operation count of this matrix multiplication is $(C^2+1)(2C+1)$ instead of $(2C+1)^3$ required for standard matrix multiplication.

The matrix multiplication $\bar{\bar{A}}_j \; \bar{\bar{P}}_{j-1} (= \bar{\bar{\alpha}}_j)$ carried out taking the advantage of sparcity as given below.

$$\underset{\bar{\bar{A}}_n}{\begin{bmatrix} \bar{\bar{I}}_C & \bar{\bar{0}}_C & \bar{0}_C \\ \bar{\bar{0}}_C & \bar{\bar{0}}_C & \bar{0}_C \\ \bar{h}_C & \bar{0} & c_{PL} \end{bmatrix}} \underset{\bar{\bar{P}}_{n-1}}{\begin{bmatrix} \bar{\bar{0}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{\bar{0}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{0}_C & \bar{X}_C & X \end{bmatrix}} = \underset{\bar{\bar{\alpha}}_n}{\begin{bmatrix} \bar{\bar{0}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{\bar{0}}_C & \bar{\bar{0}}_C & \bar{0}_C \\ \bar{0}_C & \bar{X}_C & X \end{bmatrix}} \qquad (5)$$

Here, only the $\bar{X}_C$, X in the row need to be computed.

The nonzero elements of $\bar{\bar{\alpha}}_n$ are:

$$\alpha_{j,C+i} = P_{j,C+1} \qquad \text{for } 1 \leq j \leq C+1 \qquad (6)$$
$$1 \leq i \leq C$$

$$\alpha_{2C+1,C+i} = \sum_{k=1}^{C} A_{2C+1,k} \cdot P_{k,C+i} + A_{2C+1,2C+1} \; P_{2C+1,C+i}$$

$$\text{for } \quad 1 \leq i \leq C+1 \qquad (7)$$

Thus we need to perform only $(C+1)^2$ scalar multiplication in stead of $(2C+1)^3$.

The product $\bar{\bar{A}}_n \bar{F}_{n-1} = \bar{\beta}_{n-1}$ may be found as follows

$$\underset{\bar{\bar{A}}}{\begin{bmatrix} \bar{\bar{I}}_C & \bar{\bar{0}}_C & \bar{0}_C \\ \bar{\bar{0}}_C & \bar{\bar{0}}_C & \bar{0}_C \\ \bar{h}_C & \bar{0}_C & C_{PL} \end{bmatrix}} \underset{\bar{F}}{\begin{bmatrix} \bar{F}_C \\ \bar{F}_C \\ F \end{bmatrix}} = \begin{bmatrix} \bar{F}_C \\ \bar{0}_C \\ f \end{bmatrix}$$

where $\beta_J = \bar{F}_J$ for $1 \leq J \leq C$ (8)

$$\beta_{2C+1} = \sum_{K=1}^{C} A_{2C+1,k} \quad F_k + A_{2C+1,2C+1} \quad F_{2C+1} \tag{9}$$

only the last element need to be computed; thus only C+1 operations are required instead of $(2C+1)^2$ operations for the standard matrix multiplication. In backward substitution, the multiplication $P_n \quad X_{n+1}$ involves (2C+1)(C+1) since the first C columns are zero elements.

Inversion of $(\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{P}}_{n-1})$ by partitioning.

Inversion by partitioning does not save the operations to be performed. But from the structure of the matrix $(\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{P}}_{n-1})$ some saving in operations can be realised as shown below.

$$(\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{P}}_{n-1}) = \bar{\bar{b}} = \begin{bmatrix} (1+s_n)I_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{\bar{X}}_C & \bar{\bar{X}}_C & \bar{X}_C \\ \bar{X}_C & \bar{X}_C & X \end{bmatrix}$$

$$= \begin{bmatrix} \bar{\bar{b}}_{11} & \bar{b}_{12} \\ \bar{b}_{21} & b_{22} \end{bmatrix}$$

The inverse is given by

$$\bar{\bar{b}}^{-1} = \begin{bmatrix} \bar{\bar{D}}_{11} & \bar{D}_{12} \\ \bar{D}_{21} & D_{22} \end{bmatrix}$$

where $\bar{\bar{D}}_{11} = (\bar{\bar{b}}_{11})^{-1} + (\bar{\bar{b}}_{11})^{-1} \; \bar{b}_{12} \; D_{22} \; \bar{b}_{21} \; (\bar{\bar{b}}_{11})^{-1}$

$$\bar{D}_{12} = - (\bar{\bar{b}}_{11})^{-1} \; \bar{b}_{12} \, D_{22}$$

$$\bar{D}_{21} = - D_{22} \; \bar{b}_{21} \; (\bar{\bar{b}}_{11})^{-1}$$

$$D_{22} = (b_{22} - \bar{b}_{21} (\bar{\bar{b}}_{11})^{-1} \; \bar{b}_{12})^{-1}$$

The inverse of $\bar{\bar{b}}_{11}$ can, in turn, be found by partitioning as given below.

$$\bar{\bar{b}}_{11} = \begin{bmatrix} (1+s_n) \; \bar{\bar{I}}_C & \bar{\bar{X}}_C \\ \bar{\bar{X}}_C & \bar{\bar{X}}_C \end{bmatrix} = \begin{bmatrix} \bar{\bar{d}}_{11} & \bar{\bar{d}}_{12} \\ \bar{\bar{d}}_{21} & \bar{\bar{d}}_{22} \end{bmatrix}$$

$$(\bar{\bar{b}}_{11})^{-1} = \begin{bmatrix} \bar{\bar{e}}_{11} & \bar{\bar{e}}_{12} \\ \bar{\bar{e}}_{21} & \bar{\bar{e}}_{22} \end{bmatrix}$$

where

$$\bar{\bar{e}}_{11} = (\bar{\bar{d}}_{11})^{-1} + (\bar{\bar{d}}_{11})^{-1}\,\bar{\bar{d}}_{12}\,\bar{\bar{e}}_{22}\,\bar{\bar{d}}_{21}(\bar{\bar{d}}_{11})^{-1}$$

$$\bar{\bar{e}}_{12} = -(\bar{\bar{d}}_{11})^{-1}\,\bar{\bar{d}}_{12}\,\bar{\bar{e}}_{22}$$

$$\bar{\bar{e}}_{21} = -\,\bar{\bar{e}}_{22}\,\bar{\bar{d}}_{21}(\bar{\bar{d}}_{11})^{-1}$$

$$\bar{\bar{e}}_{22} = (\bar{\bar{d}}_{22} - \bar{\bar{d}}_{21}(\bar{\bar{d}}_{11})^{-1}\,\bar{\bar{d}}_{12})^{-1}$$

It may be noted $\bar{\bar{d}}_{11}$ is a diagonal matrix and its inversion can be obtained by simply taking the reciprocal of each of the diagonal elements. Thus to find the inverse of a matrix of 2C+1, we need only the standard inversion of matrix of order C and some matrix multiplications.

	Operation Count	Standard operation	Saving
$\overset{=}{P}_j$	$(C^2+1)(2C+1)$ $= 2C^3 + C^2 + 2C+1$	$(2C+1)^3$ $= 8C^3 + 12C^2 + 6C+1$	 $6C^3 + 11C^2 + 4C$
$_j$	$(C+1)^2$ $= C^2+2C+1$	$8C^3 + 12C^2 + 6C+1$	$8C^3 + 11C^2 + 4C+1$
$_j$	$(C+1)$	$(2C+1)^2$ $= 4C^2 + 4C+1$	$4C^2 + 3C$
P_n X_n	$2C^2 + 3C+1$	$4C^2 + 4C+1$	$2C^2+C$
Inv	$5C^3 + 14C^2+6C$	$8C^3 + 12C^2 + 6C+1$	$3C^3-2C^2+1$
P_j	$(N-1)(2C^3+C^2+2C+1$	$(N-1)(8C^3+12C^2+6C+1$	
$_j$	$N (C^2 + 2C +1)$	$N (8C^3 + 12C^2 + 6C+1)$	
$_j$	$N (C + 1)$	$N (4C^2 + 4C+1)$	
$_n$ X_n	$(N-1)(C^2 + 2C+1)$	$(N-1)(4C^2 + 4 C+1)$	
Inv	$N(5C^3 + 14C^2 + 6C)$	$N (8C^3 + 12C^2 + 6C+1)$	
Total count	$3N(5C^3 + 15C^2+9C+2)$ $+ 2(N-1)(2C^3+2C^2+3C+1)$	$3N(16C^3+28C^2+16C+3)$ $+ 2(N-1)(8C^3+16C^2+10C+2)$	$3N(11C^3 + 13C^2$ $+ 7C+1)$ $+2(N-1)(6C^3 +$ $14C^2 + 7C+1)$

CHAPTER 3

RESULTS AND DISCUSSION

The algorithm proposed in the previous chapter has been implemented in FORTRAN 10 on the DEC 10 system. First, the saving in CPU time for inventing the matrix $(\bar{\bar{B}}_n - \bar{\bar{A}}_n \bar{\bar{C}}_{n-1})$ has been examined. The efficiency of the proposed method has been tested by solving two absorption problems and the details are presented in this chapter.

It is known that the inversion of matrix by partition does not result in saving the CPU time. But, the structure matrix in these separation process problems is such that we need to invert only C x C matrix to get the inversion of (2C+1) x (2C+1). This is so because the other submatrix, obtained on partitioning, whose inverse to be found to found is either identity matrix or a diagonal matrix with only a few elements which are other than one. Thus some CPU time saving can be achieved. As shown in the previous Chapter, the operation count for the inversion for a problem involving C components is $(5C^3 + 14C^2 + 6C)$ compared the standard inversion (i.e. either by the Gaussian elemination or the Gauss-Jordon elimination) which is $(2C+1)^3$.

TABLE 2

SAMPLE PROBLEM NO.1 Taken from [3]

Absorption column having 20 plates and 4 component system:

Component	Rich gas $v_{N+1,i}$ mole/hr	Absorbing liq. $l_{o,i}$ mole/hr	
A	75.0	0.0	
B	15.0	0.0	Absorbing liq. temp. = 125°C
C	10.0	0.0	Entering rich gas temp.=200°C
D	0.0	100.0	

Find the temperature and flow rates of the components of vapor and liquid stream leaving from the column.

TABLE 3

SAMPLE PROBLEM No.2

STATEMENT OF THE PROBLEM

COMPONENT	RICH GAS $v_{N+1,i}$ (mol/h)	Lean Oil $l_{o,i}$ (mol/h)	OTHER SPECIFICATIONS
CO_2	0.4703	0.0	T_o= 2.9°F, T_{N+1}=0°F,
N_2	0.1822	0.0	N = 8, and
CH_4	88.7000	0.0	P = 800 lb/in^2 abs
C_2H_6	6.6747	0.0	initial temperature
C_3H_8	2.7786	0.0015	profile to be constant
i C_4H_{10}	0.6375	0.0006	at T_j = 25°F
n C_4H_{10}	0.3655	0.0013	for all j (j = 1,2...N).
i C_5H_{12}	0.1158	0.0067	The initial vapor rate
n C_5H_{12}	0.0505	0.0061	profile is to be constant
C_6H_{14}	0.0146	0.1495	at V_j= 90.88(j=1,2...8)
C_7H_{16}	0.0081	0.5736	and the liquid rates are
C_8H_{18}	0.002	1.8214	L_j = 6.3095 (j = 1,2...8)
C_9H_{20}	0.000	1.6866	and L_8 = 15.42
$C_{10}H_{22}$	0.000	2.0619	

TABLE 5

INITIAL GUESS OF FLOW RATES AND TEMPERATURE ARE:

VAPOR PHASE COMPONENT FLOW RATES ARE

COMPONENT NO.	1	2	3	4	5	6	7	8
1	[illegible]	.4241E+00	.4240E+00	.4240E+00	.4241E+00	.4242E+00	.4243E+00	.4248E+00
2	[illegible]	.1643E+00	.1643E+00	.1643E+00	.1643E+00	.1643E+00	.1644E+00	.1646E+00
3	[illegible]	[illegible]	.7997E+02	.7996E+02	.7998E+02	.8000E+02	.8003E+02	.8012E+02
4	[illegible]	[illegible]	.6018E+01	.6018E+01	.6019E+01	.6020E+01	.6022E+01	.6029E+01
5	[illegible]	.2499E+01	.2505E+01	.2505E+01	.2506E+01	.2506E+01	.2507E+01	.2510E+01
6	[illegible]	.5679E+00	.5740E+00	.5747E+00	.5748E+00	.5750E+00	.5752E+00	.5758E+00
7	.2765E+00	.3206E+00	.3281E+00	.3293E+00	.3295E+00	.3296E+00	.3298E+00	.3301E+00
8	.7573E-01	.9555E-01	.1011E+00	.1033E+00	.1040E+00	.1043E+00	.1044E+00	.1046E+00
9	[illegible]	.3977E-01	.4216E-01	.4396E-01	.4484E-01	.4524E-01	.4544E-01	.4557E-01
10	[illegible]	.1017E+00	.9891E-01	.8675E-01	.7694E-01	.6513E-01	.5092E-01	.3383E-01
11	[illegible]	.1730E+00	.1728E+00	.1724E+00	.1710E+00	.1669E+00	.1546E+00	.1178E+00
12	.2417E+00	.2400E+00	.2399E+00	.2399E+00	.2398E+00	.2392E+00	.2350E+00	.2055E+00
13	.1823E+00	.1810E+00	.1809E+00	.1809E+00	.1809E+00	.1807E+00	.1785E+00	.1597E+00
14	[illegible]	.6281E-01	.6179E-01	.6179E-01	.6180E-01	.6180E-01	.6176E-01	.5984E-01

LIQ. COMP. FLOW RATES ARE

COMPONENT NO.	1	2	3	4	5	6	7	8
1	[illegible]	.6403E-02	.6407E-02	.6417E-02	.6433E-02	.6466E-02	.6575E-02	.1764E-01
2	.9732E-04	.9746E-04	.9751E-04	.9764E-04	.9789E-04	.9840E-04	.1001E-03	.2684E-03
3	.3057E+00	.3056E+00	.3057E+00	.3061E+00	.3069E+00	.3085E+00	.3137E+00	.8415E+00
4	[illegible]	.1058E+00	.1059E+00	.1060E+00	.1062E+00	.1068E+00	.1086E+00	.2914E+00
5	.1279E+00	.1343E+00	.1347E+00	.1349E+00	.1352E+00	.1359E+00	.1382E+00	.3708E+00
6	.5897E-01	.6498E-01	.6564E-01	.6582E-01	.6599E-01	.6634E-01	.6745E-01	.1809E+00
7	.4523E-01	.5226E-01	.5339E-01	.5377E-01	.5394E-01	.5422E-01	.5513E-01	.1479E+00
8	.2525E-01	.3156E-01	.3375E-01	.3453E-01	.3488E-01	.3515E-01	.3577E-01	.9598E-01
9	.1379E-01	.1755E-01	.1939E-01	.2025E-01	.2069E-01	.2098E-01	.2142E-01	.5758E-01
10	[illegible]	.1209E+00	.1128E+00	.1033E+00	.9183E-01	.7812E-01	.6208E-01	.1105E+00
11	.5129E+00	.5105E+00	.5102E+00	.5096E+00	.5068E+00	.4972E+00	.4681E+00	.9558E+00
12	.1620E+01	.1622E+01	.1622E+01	.1625E+01	.1628E+01	.1632E+01	.1630E+01	.3819E+01
13	[illegible]	.1502E+01	.1502E+01	.1504E+01	.1508E+01	.1514E+01	.1520E+01	.3643E+01
14	[illegible]	.1831E+01	.1837E+01	.1839E+01	.1844E+01	.1853E+01	.1883E+01	.4888E+01

Table 5 (continued)

```
INITIAL VAPOR AND LIQUID RATE AND  TEMPERATURE GUESS ARE

---------------------------------------------------------
STAGE NUMBER      VAPOR RATE           LIQUID RATE         TEMPERATURE
------------      ------------------   -----------------   -----------------

      1            0.90880000E+02      0.63092000E+01      0.48500000E+03
      2            0.90880000E+02      0.63092000E+01      0.48500000E+03
      3            0.90880000E+02      0.63092000E+01      0.48500000E+03
      4            0.90880000E+02      0.63092000E+01      0.48500000E+03
      5            0.90880000E+02      0.63092000E+01      0.48500000E+03
      6            0.90880000E+02      0.63092000E+01      0.48500000E+03
      7            0.90880000E+02      0.63092000E+01      0.48500000E+03
      8            0.90880000E+02      0.15420000E+02      0.48500000E+03

            COMPONENT        TOP LIQ.FEED          BOTTOM VAP.FEED.
            ----------     -----------------     -----------------
                1           0.00000000E+00         0.47030000E+00

                2           0.00000000E+00         0.18220000E+00

                3           0.00000000E+00         0.88700000E+02

                4           0.00000000E+00         0.66747000E+01

                5           0.15000000E-02         0.27786000E+01

                6           0.80000000E-02         0.63750000E+00

                7           0.13000000E-02         0.36550000E+00

                8           0.67000000E-02         0.11580000E+00

                9           0.81000000E-02         0.50500000E-01

               10           0.14950000E+00         0.14600000E-01

               11           0.57360000E+00         0.81000000E-02

               12           0.18214000E+01         0.29000000E-02

               13           0.16585000E+01         0.00000000E+00

               14           0.20610000E+01         0.00000000E+00
```

Table 5 (continued)

```
     APP AFTER MASS BALANCE= 0.8625E+02
     J=  5      ERROR ENTHALPY = 0.1872E+11

ITERATION NUMBER=  1

     APP AFTER MASS BALANCE= 0.2946E-11
     J=  5      ERROR ENTHALPY = 0.6312E+06

ITERATION NUMBER=  2

     APP AFTER MASS BALANCE= 0.2471E-11
     J=  5      ERROR ENTHALPY = 0.7279E+05

ITERATION NUMBER=  3

     APP AFTER MASS BALANCE= 0.2617E-11
     J=  5      ERROR ENTHALPY = 0.5145E+03

ITERATION NUMBER=  4

     APP AFTER MASS BALANCE= 0.1765E-11
     J=  5      ERROR ENTHALPY = 0.8651E-03

ITERATION NUMBER=  5

     APP AFTER MASS BALANCE= 0.4281E-11
     J=  5      ERROR ENTHALPY = 0.5946E-03

SUCCESSFUL CONVERGENCE

ITERATION NUMBER=  5
```

Table 5 (continued)

```
FINAL VALUES  OF FLOW RATES AND TEMPERATURE ARE:

VAPOR PHASE COMPONENT FLOW RATES ARE
-------------------------------------
COMPONENT NO.            STAGE NOS. ARE
-------------            --------------

                 1            2            3            4            5            6            7            8
----------------------------------------------------------------------------------------------------------------
 1          .3620E+00    .4149E+00    .4198E+00    .4220E+00    .4247E+00    .4284E+00    .4340E+00    .4439E+00
 2          .1805E+00    .1815E+00    .1815E+00    .1815E+00    .1816E+00    .1816E+00    .1817E+00    .1818E+00
 3          .8337E+02    .8664E+02    .8664E+02    .8671E+02    .8683E+02    .8698E+02    .8721E+02    .8762E+02
 4          .4905E+01    .5734E+01    .5827E+01    .5867E+01    .5914E+01    .5977E+01    .6072E+01    .6238E+01
 5          .8805E+00    .1326E+01    .1517E+01    .1629E+01    .1720E+01    .1818E+01    .1951E+01    .2181E+01
 6          .2675E-01    .4921E-01    .7027E-01    .9344E-01    .1218E+00    .1605E+00    .2202E+00    .3319E+00
 7          .2617E-02    .5222E-02    .8704E-02    .1396E-01    .2255E-01    .3776E-01    .6768E-01    .1370E+00
 8          .2216E-02    .2310E-02    .2231E-02    .2164E-02    .2269E-02    .3040E-02    .6469E-02    .2210E-01
 9          .1487E-02    .1563E-02    .1497E-02    .1408E-02    .1331E-02    .1373E-02    .2122E-02    .7328E-02
10          .1409E-01    .1494E-01    .1443E-01    .1365E-01    .1270E-01    .1149E-01    .9855E-02    .7703E-02
11          .2182E-01    .2321E-01    .2249E-01    .2135E-01    .1993E-01    .1814E-01    .1565E-01    .1156E-01
12          .3017E-01    .3211E-01    .3115E-01    .2962E-01    .2771E-01    .2529E-01    .2191E-01    .1617E-01
13          .2266E-01    .2410E-01    .2339E-01    .2225E-01    .2084E-01    .1904E-01    .1653E-01    .1223E-01
14          .7680E-02    .8162E-02    .7926E-02    .7552E-02    .7084E-02    .6491E-02    .5657E-02    .4227E-02
----------------------------------------------------------------------------------------------------------------

LIQ. COMP. FLOW RATES ARE
-------------------------------------
COMPONENT NO.            STAGE NOS. ARE
-------------            --------------

                 1            2            3            4            5            6            7            8
----------------------------------------------------------------------------------------------------------------
 1          .5284E-01    .5773E-01    .5992E-01    .6265E-01    .6639E-01    .7197E-01    .8188E-01    .1083E+00
 2          .1067E-02    .1026E-02    .1052E-02    .1092E-02    .1146E-02    .1225E-02    .1363E-02    .1725E-02
 3          .3264E+01    .3264E+01    .3311E+01    .3455E+01    .3611E+01    .3839E+01    .4245E+01    .5328E+01
 4          .6291E+00    .9290E+00    .9619E+00    .1009E+01    .1073E+01    .1167E+01    .1333E+01    .1770E+01
 5          .4468E+00    .6321E+00    .7501E+00    .8407E+00    .9387E+00    .1072E+01    .1302E+01    .1900E+01
 6          .2846E-01    .4951E-01    .7268E-01    .1011E+00    .1397E+00    .1995E+00    .3111E+00    .6167E+00
 7          .3915E-02    .7397E-02    .1254E-01    .2123E-01    .3645E-01    .6636E-01    .1357E+00    .3642E+00
 8          .5872E-02    .6715E-02    .6612E-02    .6753E-02    .7523E-02    .1095E-01    .2659E-01    .1203E+00
 9          .6176E-02    .5109E-02    .5021E-02    .5943E-02    .5985E-02    .6735E-02    .1194E-01    .5511E-01
10          .1503E+00    .1498E+00    .1491E+00    .1481E+00    .1469E+00    .1453E+00    .1431E+00    .1500E+00
11          .5757E+00    .5743E+00    .5731E+00    .5717E+00    .5699E+00    .5674E+00    .5633E+00    .5599E+00
12          .1823E+01    .1822E+01    .1821E+01    .1819E+01    .1817E+01    .1813E+01    .1807E+01    .1793E+01
13          .1688E+01    .1687E+01    .1686E+01    .1685E+01    .1683E+01    .1680E+01    .1676E+01    .1664E+01
14          .2062E+01    .2062E+01    .2062E+01    .2061E+01    .2061E+01    .2060E+01    .2058E+01    .2054E+01
----------------------------------------------------------------------------------------------------------------
```

Table 5 (continued)

VAPOR AND LIQUID RATE AND TEMPERATURE ARE

STAGE NUMBER	VAPOR RATE	LIQUID RATE	TEMPERATURE
1	0.89829526E+02	0.10938037E+02	0.50106360E+03
2	0.94452963E+02	0.11248682E+02	0.50395584E+03
3	0.94763607E+02	0.11502861E+02	0.50320764E+03
4	0.95017786E+02	0.11788229E+02	0.50124028E+03
5	0.95323155E+02	0.12156413E+02	0.49855777E+03
6	0.95671336E+02	0.12701652E+02	0.49504246E+03
7	0.96215577E+02	0.13696913E+02	0.49003263E+03
8	0.97211838E+02	0.16484876E+02	0.48148252E+03

OUTPUT OF PROB NO. 1

TABLE 6

```
 STAGE NUMBER    COMPONENT NO.    COMP VAP RATE    COMPONENT LIQ RATE
 ------------    -------------   ---------------   ------------------
       1               1          0.7484947E+02     0.2100884E+00
       1               2          0.5563007E+01     0.4943348E+01
       1               3          0.1110582E+00     0.1634238E+00
       1               4          0.8543292E-04     0.1000000E+03
       2               1          0.7505955E+02     0.1329435E+00
       2               2          0.1050635E+02     0.5704169E+01
       2               3          0.2744820E+00     0.2841667E+00
       2               4          0.1343854E-03     0.1000000E+03
       3               1          0.7498241E+02     0.1310542E+00
       3               2          0.1126718E+02     0.6315327E+01
       3               3          0.3952249E+00     0.4150736E+00
       3               4          0.1310780E-03     0.1000000E+03
       4               1          0.7498052E+02     0.1296111E+00
       4               2          0.1187833E+02     0.6683225E+01
       4               3          0.5261319E+00     0.5412112E+00
       4               4          0.1311656E-03     0.1000000E+03
       5               1          0.7497908E+02     0.1380885E+00
       5               2          0.1224623E+02     0.7281966E+01
       5               3          0.6522694E+00     0.6937689E+00
       5               4          0.1254914E-03     0.1000000E+03
       6               1          0.7498755E+02     0.1388031E+00
       6               2          0.1284497E+02     0.7651230E+01
       6               3          0.8048271E+00     0.8449997E+00
       6               4          0.1259466E-03     0.1000000E+03
       7               1          0.7498827E+02     0.1459628E+00
       7               2          0.1321424E+02     0.8213796E+01
       7               3          0.9560579E+00     0.1039321E+01
       7               4          0.1220132E-03     0.1000000E+03
       8               1          0.7499543E+02     0.1499273E+00
       8               2          0.1377680E+02     0.8747403E+01
       8               3          0.1150379E+01     0.1273688E+01
       8               4          0.1205397E-03     0.1000000E+03
       9               1          0.7499939E+02     0.1530168E+00
       9               2          0.1431041E+02     0.9232110E+01
       9               3          0.1384746E+01     0.1557057E+01
       9               4          0.1196637E-03     0.1000000E+03
      10               1          0.7500248E+02     0.1590006E+00
      10               2          0.1479512E+02     0.9858971E+01
      10               3          0.1668115E+01     0.1948527E+01
      10               4          0.1172821E-03     0.1000000E+03
---------------------------------------------------------------
```

Table 6 (continued)

STAGE NUMBER	COMPONENT NO.	COMP. VAP RATE	COMPONENT LIQ RATE
11	1	0.7500847E+02	0.1580991E+00
11	2	0.1542198E+02	0.1019131E+02
11	3	0.2059585E+01	0.2389557E+01
11	4	0.1189466E-03	0.1000000E+03
12	1	0.7500756E+02	0.1588155E+00
12	2	0.1575432E+02	0.1042882E+02
12	3	0.2500615E+01	0.2913677E+01
12	4	0.1195391E-03	0.1000000E+03
13	1	0.7500828E+02	0.1588129E+00
13	2	0.1599182E+02	0.1056358E+02
13	3	0.3024736E+01	0.3524292E+01
13	4	0.1205190E-03	0.1000000E+03
14	1	0.7500828E+02	0.1586043E+00
14	2	0.1612659E+02	0.1061237E+02
14	3	0.3635350E+01	0.4229188E+01
14	4	0.1215733E-03	0.1000000E+03
15	1	0.7500807E+02	0.1578280E+00
15	2	0.1617538E+02	0.1058948E+02
15	3	0.4340246E+01	0.5022740E+01
15	4	0.1231780E-03	0.1000000E+03
16	1	0.7500729E+02	0.1566930E+00
16	2	0.1613249E+02	0.1044457E+02
16	3	0.5133798E+01	0.5897304E+01
16	4	0.1249106E-03	0.1000000E+03
17	1	0.7500616E+02	0.1552666E+00
17	2	0.1600758E+02	0.1025007E+02
17	3	0.6008362E+01	0.6838635E+01
17	4	0.1268369E-03	0.1000000E+03
18	1	0.7500473E+02	0.1536361E+00
18	2	0.1581307E+02	0.1000183E+02
18	3	0.6949693E+01	0.7828102E+01
18	4	0.1288986E-03	0.1000000E+03
19	1	0.7500310E+02	0.1520045E+00
19	2	0.1556483E+02	0.9722459E+01
19	3	0.7939161E+01	0.8850086E+01
19	4	0.1309856E-03	0.1000000E+03
20	1	0.7500147E+02	0.1505351E+00
20	2	0.1528547E+02	0.9436994E+01
20	3	0.8961144E+01	0.9888942E+01
20	4	0.1330168E-03	0.9999991E+02

Table 6 (continued)

```
   STAGE NO.     TOTAL LIQ. RATE     TOTAL VAP. RATE     TEMPERATURE
   ---------     ---------------     ---------------     -----------
         1        0.10531691E+03      0.80523616E+02      0.14468347E+03
         2        0.10612133E+03      0.85840525E+02      0.14783305E+03
         3        0.10686150E+03      0.86644940E+02      0.15064070E+03
         4        0.10735409E+03      0.87385118E+02      0.15253474E+03
         5        0.10811385E+03      0.87877704E+02      0.15540042E+03
         6        0.10863507E+03      0.88637480E+02      0.15741018E+03
         7        0.10939911E+03      0.89158686E+02      0.16031813E+03
         8        0.11017105E+03      0.89922730E+02      0.16326849E+03
         9        0.11094221E+03      0.90694669E+02      0.16623979E+03
        10        0.11196653E+03      0.91465831E+02      0.17015918E+03
        11        0.11273900E+03      0.92490149E+02      0.17316539E+03
        12        0.11350135E+03      0.93262614E+02      0.17615109E+03
        13        0.11424672E+03      0.94024961E+02      0.17909611E+03
        14        0.11500020E+03      0.94770336E+02      0.18209372E+03
        15        0.11575009E+03      0.95523818E+02      0.18509986E+03
        16        0.11649861E+03      0.95273704E+02      0.18812043E+03
        17        0.11724401E+03      0.97022224E+02      0.19114513E+03
        18        0.11798361E+03      0.97767625E+02      0.19415974E+03
        19        0.11872460E+03      0.98507227E+02      0.19718606E+03
        20        0.11947639E+03      0.99248212E+02      0.20019488E+03
   AEF AFTER MASS BALANCE= 0.1458E+03
   AEF AFTER MASS BALANCE= 0.1461E+03
   AEF AFTER MASS BALANCE= 0.1462E+03
   AEF AFTER MASS BALANCE= 0.1981E+03
   J= 20     AEFOF ENTHALPY = 0.6342E+03

ITERATION NUMBER=  1
   AEF AFTER MASS BALANCE= 0.9920E-11
   AEF AFTER MASS BALANCE= 0.9920E-11
   AEF AFTER MASS BALANCE= 0.9920E-11
   AEF AFTER MASS BALANCE= 0.1083E-10
   J= 20     AEFOF ENTHALPY = 0.3438E-03

ITERATION NUMBER=  2
   AEF AFTER MASS BALANCE= 0.2093E-10
   AEF AFTER MASS BALANCE= 0.2095E-10
   AEF AFTER MASS BALANCE= 0.2096E-10
   AEF AFTER MASS BALANCE= 0.2096E-10
   J= 20     AEFOF ENTHALPY = 0.2143E-09

SUCESSFUL CONVERGENT

ITERATION NUMBER=  2
```

To get actual CPU time savings, numerical experiments were carried out using the test matrix of the type that are encountered in the separation process problems. The test matrices are tabulated in Appendix (D) . The CPU time required for the inversion by partition and by the Gauss-Jordon elemination have been compared for different values of C and presented in Table 1.

The saving in CPU time is becoming increasingly significant as the C increases if the proposed method is employed.

An absorption problem given by Naphthali-Sandholm 3 and another absorption problem given by Holland 1 have been chosen for testing the efficiency of the proposed method. In the first problem, the total number components of the mixture are four and the number stages are twenty. In the other problem the number components are fourteen and stages are eight. The details of the problems are given in Table 2 and 3 and the equilibrium and enthalpy data are given in Appendices B & C. The elements of the Jocabian matrix (the partial derivatives) have been evaluated analytically. The analytical expressions for the derivatives are given in Appendix (A & E).

The two problems have been solved exploiting the sparsity of the submatrices and employing the 'standard' matrix operations. Since no approximations were made while taking of the sparsity in computation, the number of iteration required identical in both the methods. The CPU time required to solve the two problems by these two methods alongwith other details are given in Table 4.

Table 4 shows a substantial reduction in CPU time is achieved in the proposed method. The saving in CPU time becomes increasingly significant as the number of components increases. The proposed method can be employed even interlinked columns.

CHAPTER 4

CONCLUSIONS

An efficient algorithm for solving the separation processes problems by the well known Naphthali-Sandholm method has been presented. In this algorithm the sparsity the submatrices of the Jacobian matrix is exploited in the matrix multiplications and in the inversion of the matrices $(\bar{\bar{B}}_n - \bar{A}_n \bar{\bar{C}}_{n-1})$. It has been shown that the operation count for the proposed algorithm is $(5C^3 + 14C^2 + 6C)$ compared to $(2C+1)^3$ with the standard matrix operations. By solving two 'bench-mark' problems, it has been shown that the saving in the CPU time becomes increasingly significant as the number components involved becomes large. In the proposed algorithm, the saving in CPU time in computing the correction vector $\Delta \bar{X}$ is effected, but the number of iteration required is the same as with the use of standard matrix operations.

The algorithm can be extended for solving interlinked column employing the method suggested by Hocling and Seader. The computer code need to be tested to determine its effectiveness for distillation of nonideal mixtures, extraction and adsorption problems.

NOMENCLATURE

$\overset{=}{A}_j$	Submatrices of Jacobian Matrix at the jth row
$\overset{=}{B}_j$	Submatrices of Jacobian Matrix at the jth row
$\overset{=}{C}_j$	Submatrices of Jacobian Matrix at the jth row
C	Total number of components involved
L_j	Overall molar liquid flow rate from jth stage
V_j	Overall molar Vapor flow rate from jth stage
l_{ij}	Molar liquid flow rate of component i from jth stage
v_{ij}	Molar vaprr flow rate of component i from jth stage
K_{ij}	Distribution coefficient of ith component at jth stage
$\overline{X}$	Vector of variables
$\overset{=}{J}$	Jacobian matrix
f_{ij}	Molar feed rate of the component i into the jth stage
$\overline{F}$	Residual vector
$\overline{F}_j$	Residual vector for the jth stage
$\overset{=}{P}_j$	P matrix in the Thomas algorithm
q_j	External heat input into the jth stage
h_i	Molar specific liquid phase enthalpy of component i
H_i	Molar specific vapor phase enthalpy of component i
$h_{f_{i,j}}$	Molar specific feed enthalpies of component i
N	Total number of stages
M_{ij}	Mass balance discrepancy of component i for jth stage

O_{ij}	Equilibrium relation discrepancy function for component i at jth stage
E_j	Enthalpy balance discrepancy function for the jth stage
S_j	Fraction of the vapor stream withdrawn from the jth stage
s_j	Fraction of the liquid stream withdrawn from the jth stage

Greek letters

Δ	difference in variable
η_j	murphree efficiency of jth stage
δ_{ij}	Kronecker delta function
∂	partial derivative
γ_i	activity coefficient of component i
Λ	Thermodynamic parameter
$\bar{\bar{\alpha}}$	P matrix of the Thomas Algorithm
$\bar{\beta}$	q matrix in the Thomas Algorithm

REFERENCES

1. Hofeling, B.S., and Seader, J.D., AIChE J. 24, 1131 (1978)
2. Holland, C.D., "Fundamental of multicomponent Distillation", McGraw Hill Inc. (1981)
3. Ketchum, R.G., Chem. Engg. Sci., 34, 387 (1979)
4. Naphthali, L.M., 56th AIChE meeting, San Fransisco, May (1965)
5. Naphthali, L.M., and Sandholm, D.P., AIChE J. 17, 1 (1971)
6. Stadtherr, M.A., AIChE J., 25, 4, 609 (1979)
7. Stadtherr, M.A., and Wood, E.S., Comput. Chem. Engng., 6, 115 (1980)
8. Stadtherr, M.A. and Hilten, M.C., Comput. Chem. Engng., 6, 121 (1982).

TABLE 1

CPU time for the inversion of matrix by the proposed method and the Gauss Jordan Elimination

No. of components	CPU time (in Secs.)		Ratio of CPU time (IP/GJ)	Ratio of operation count
	Inversion by partition(I.P)	Gauss Jordan Elimination (GJ)		
5	0.14	0.25	0.56	0.76
10	0.59	1.38	0.43	0.70
15	1.96	5.54	0.35	0.68
20	4.99	16.22	0.31	0.66

TABLE 4

CPU time for two different problems

Problem No.	No. of components	No. of stages	No. of iterations needed I	Run No.	CPU in secs.		Ratio of CPU per iteration
					Normal t_1	Symmetry sparcity exploitation t_2	$\frac{t_2}{I\ t_1}$
1	4	20	2	1	7.95	1.95	0.12
				2	7.93	1.96	0.12
				3	7.95	1.96	0.12
2	14	8	5	1	784.64	57.12	0.01
				2	784.62	57.11	0.01
				3	784.62	57.14	0.01

APPENDIX A

ELEMENTS OF THE SUBMATRICES

1. FOR IDEAL VLE SYSTEM

A. FOR $\overset{=}{A}$ MATRIX

For $1 \leq i,\ j \leq C$ & $1 \leq n \leq N$

$$\frac{\partial M_{i,n}}{\partial l_{j,n-1}} = \delta_{i,j} \ , \quad \frac{\partial M_{i,n-}}{\partial v_{j,n-1}} = \frac{\partial M_{i,n}}{\partial T_{n-1}} = 0$$

$$\frac{\partial O_{i,n}}{\partial l_{j,n-1}} = \frac{\partial O_{i,n}}{\partial v_{j,n-1}} = \frac{\partial O_{i,n}}{\partial T_{n-1}} = 0$$

$$\frac{\partial E_n}{\partial l_{i,n-1}} = h_{i,n-1} \ ; \quad \frac{\partial E_n}{\partial v_{i,n-1}} = 0$$

$$\frac{\partial E_n}{\partial T_n} = \sum_{i=1}^{C} l_{i,n-1} \ \frac{\partial h_{i,n-1}}{\partial T_{n-1}}$$

B. FOR $\overset{=}{B}$ MATRIX

For $1 \leq i,\ j \leq C$ & $1 \leq n \leq N$

$$\frac{\partial M_{i,n}}{\partial l_{j,n}} = -(1+ s_n)\delta_{ij}$$

$$\frac{\partial M_{i,n}}{\partial v_{j,n}} = -(1+ S_n)\delta_{ij}$$

$$\frac{\partial M_{i,n}}{\partial T_n} = 0$$

Appendix A continued.

$$\frac{\partial o_{i,n}}{\partial l_{j,n}} = \eta_n \, k_{i,n} \quad \frac{\delta_{ij} L_n - l_{in}}{L_n^2}$$

$$\frac{\partial o_{i,n}}{\partial v_{j,n}} = \quad \frac{v_{i,n} - V_n \delta_{ij}}{V_n^2}$$

$$\frac{\partial o_{i,n}}{\partial T_n} = \frac{\eta_n \, l_{i,n}}{L_n} \quad \frac{dk_{i,n}}{dT_n}$$

$$\frac{\partial E_n}{\partial l_{i,n}} = -\,(1+ s_m)\, h_{i,n}$$

$$\frac{\partial E_n}{\partial v_{i,n}} = -\,(1+ S_m)\, H_{i,n}$$

$$\frac{\partial E_n}{\partial T_n} = -\sum_{i=1}^{c} (1+S_m)\, v_{i,n} \frac{dH_{i,n}}{dT_m} - \sum_{i=1}^{c} (1+s_n)\, l_{i,n} \frac{dh_{i,n}}{dT_n}$$

C. FOR $\overset{=}{C}$ MATRIX

For 1 i,j C & 1 n N

$$\frac{\partial M_{i,n}}{\partial l_{j,n+1}} = 0; \quad \frac{\partial M_{i,n}}{\partial v_{j,n+1}} = \delta_{ij} \; ; \quad \frac{\partial M_{i,n}}{\partial T_{n+1}} = 0$$

$$\frac{\partial o_{i,n}}{\partial l_{j,n+1}} = 0 \; ; \quad \frac{\partial o_{i,n}}{\partial v_{j,n+1}} = (1-\eta_m) \quad \frac{\delta_{ij} V_{n+1} - v_{i,n+1}}{V_{n+1}^2}$$

Appendix A continued

$$\frac{\partial O_{i,n}}{\partial T_{n+1}} = 0 = \frac{\partial E_n}{\partial I_{i,n+1}}$$

$$\frac{\partial E_n}{\partial v_{i,n+1}} = H_{i,n+1} \quad ; \quad \frac{\partial E_n}{\partial T_{n+1}} = \sum_{i=1}^{c} v_{i,n+1} \frac{dH_{i,n+1}}{dT_{n+1}}$$

APPENDIX B

SAMPLE PROBLEM No.1

DATA

K values

Material	Temperature	
	100°F	200°F
A	500.0	550.0
B	1.50	1.8
C	0.90	1.00
D	1.0×10^{-6}	1.5×10^{-6}
	Molar liquid enthalpies, 10^3 cal/mole.	
A	0.01	0.013
B	0.30	0.33
C	0.40	0.44
D	1.50	1.90
	Molar vapor enthalpies, 10^3 cal/mole	
A	1.00	1.002
B	1.80	1.82
C	2.00	2.03
D	5.75	5.95

K values for the temperature range of - 25°F to 40°F and at a pressure of 800 lb/in^{2}abs.

Component	a_{1i}	a_{2i}	a_{3i}	a_{4i}
CO_2	$- 0.62822223 \times 10^{-1}$	$0.30688802 \times 10^{-3}$	$0.39996468 \times 10^{-6}$	$-0.57899830 \times 10^{-9}$
N_2	0.50596821	$-0.43488364 \times 10^{-3}$	$-0.15009991 \times 10^{-5}$	$0.34494154 \times 10^{-8}$
CH_4	0.15584934	$-0.15205775 \times 10^{-3}$	$0.50349212 \times 10^{-6}$	$-0.17713546 \times 10^{-9}$
C_2H_6	$0.91486037 \times 10^{-1}$	$-0.16355944 \times 10^{-3}$	$0.33741924 \times 10^{-6}$	$0.14797150 \times 10^{-9}$
C_3H_8	$0.37769508 \times 10^{-1}$	$-0.64491702 \times 10^{-4}$	$0.29233627 \times 10^{-6}$	$-0.48597680 \times 10^{-11}$
i C_4H_{10}	$0.36708355 \times 10^{-1}$	$-0.94310963 \times 10^{-4}$	$0.28026648 \times 10^{-6}$	$0.10462797 \times 10^{-10}$
n C_4H_{10}	$0.37231278 \times 10^{-1}$	$-0.13635085 \times 10^{-3}$	$0.37584653 \times 10^{-6}$	$-0.69237741 \times 10^{-10}$
n C_5H_{12}	$0.15414596 \times 10^{-1}$	$-0.34736106 \times 10^{-4}$	$0.12591028 \times 10^{-6}$	$0.73157133 \times 10^{-10}$
i C_5H_{12}	$0.19747034 \times 10^{-1}$	$-0.40284984 \times 10^{-4}$	$0.14439195 \times 10^{-6}$	$0.56656790 \times 10^{-10}$
n C_6H_{14}	$0.88765752 \times 10^{-3}$	$0.37082646 \times 10^{-4}$	$-0.40746951 \times 10^{-7}$	$0.15187203 \times 10^{-9}$
n C_7H_{16}	$0.63677356 \times 10^{-2}$	$-0.64409760 \times 10^{-5}$	$0.31793974 \times 10^{-7}$	$0.78284379 \times 10^{-10}$
n C_8H_{18}	$0.99674799 \times 10^{-2}$	$-0.34673591 \times 10^{-4}$	$0.82305291 \times 10^{-7}$	$0.21022392 \times 10^{-10}$
n C_9H_{20}	$0.78793392 \times 10^{-2}$	$-0.23886125 \times 10^{-4}$	$0.62435951 \times 10^{-7}$	$0.25793478 \times 10^{-10}$
n $C_{10}H_{22}$	$0.64146556 \times 10^{-2}$	$-0.16131104 \times 10^{-4}$	$0.30005250 \times 10^{-7}$	$0.30266026 \times 10^{-10}$

$$K_i = T \left(a_{1i} + a_{2i} T + a_{3i} T^2 + a_{4i} T^3 \right)^3 \quad (T \text{ in } {}^{\circ}R)$$

LIQUID ENTHALPIES FOR THE TEMPERATURE RANGE OF - 25°F to 40°F at P = 800 lb/in^{2}abs.

Component	b_{1i}	b_{2i}	b_{3i}	b_{4i}
CO_2	$0.22524075\text{x}10^4$	$0.5446243\text{x}10^1$	$0.2791080\text{x}10^{-1}$	$-0.18765335\text{x}10^{-4}$
N_2	$0.15837112\text{x}10^4$	$0.3731512\text{x}10^1$	$0.17655857\text{x}10^{-1}$	$-0.14662071\text{x}10^{-4}$
CH_4	$0.81635181\text{x}10^3$	$0.7206460\text{x}10^1$	$0.15354034\text{x}10^{-1}$	$-0.84406456\text{x}10^{-5}$
C_2H_6	$0.97404712\text{x}10^3$	$0.11454294\text{x}10^2$	$0.79399594\text{x}10^{-2}$	$-0.42183183\text{x}10^{-6}$
C_3H_8	$0.21237510\text{x}10^4$	$0.46383524\text{x}10^1$	$0.31726830\text{x}10^{-1}$	$-0.12580301\text{x}10^{-4}$
i C_4H_{10}	$0.17543628\text{x}10^4$	$0.92456856\text{x}10^1$	$0.30206113\text{x}10^{-1}$	$-0.89584664\text{x}10^{-5}$
n C_4H_{10}	$0.32309192\text{x}10^4$	$0.66175545\text{x}10^1$	$0.38262386\text{x}10^{-1}$	$-0.16110935\text{x}10^{-4}$
i C_5H_{12}	$0.33611663\text{x}10^4$	$0.39552670\text{x}10^1$	$0.54925641\text{x}10^{-1}$	$-0.25869682\text{x}10^{-4}$
n C_5H_{12}	$0.43454375\text{x}10^4$	$0.10596339\text{x}10^2$	$0.43731511\text{x}10^{-1}$	$-0.19637475\text{x}10^{-4}$
n C_6H_{14}	$-0.44150469\text{x}10^4$	$0.70354599\text{x}10^2$	$-0.67470074\text{x}10^{-1}$	$0.60245657\text{x}10^{-4}$
n C_7H_{16}	$0.66707016\text{x}10^2$	$0.18159073\text{x}10^2$	$0.38164884\text{x}10^{-1}$	$-0.42837073\text{x}10^{-5}$
n C_8H_{18}	$-0.10632578\text{x}10^2$	$0.19229950\text{x}10^2$	$0.40186413\text{x}10^{-1}$	$-0.70521889\text{x}10^{-6}$
n C_9H_{20}	$-0.79141992\text{x}10^4$	$0.81615143\text{x}10^2$	$-0.79501927\text{x}10^{-1}$	$0.83943509\text{x}10^{-4}$
n $C_{10}H_{22}$	$-0.67810352\text{x}10^4$	$0.74108551\text{x}10^2$	$-0.58315706\text{x}10^{-1}$	$0.75087155\text{x}10^{-4}$

$h_i = b_{1i} + b_{2i}T + b_{3i}T^2 + b_{4i}T^3$ (T in °R) Btu/lb mole.

Appendix C continued.

VAPOR ENTHALPIES FOR THE TEMPERATURE RANGE OF - 25°F To 40°F AT P = 800 lb/in^{2}abs.

Component	C_{1i}	C_{2i}	C_{3i}	C_{4i}
CO_2	$0.13978977x10^5$	$- 0.96359463x10^1$	$0.38228422x10^{-1}$	$-0.26870170x10^{-4}$
N_2	$0.48638672x10^4$	$- 0.21227379x10^1$	$0.17565668x10^{-1}$	$-0.11367006x10^{-4}$
CH_4	$0.63255430x10^4$	$- 0.20747757x10^1$	$0.18532634x10^{-1}$	$-0.10630416x10^{-4}$
C_2H_6	$0.10628934x10^5$	$- 0.28718834x10^1$	$0.24877094x10^{-1}$	$-0.13233222x10^{-4}$
C_3H_8	$0.13954383x10^5$	$- 0.41930256x10^1$	$0.32614145x10^{-1}$	$-0.15483340x10^{-4}$
i C_4H_{10}	$0.94088984x10^4$	$0.39262680x10^2$	$-0.55596594x10^{-1}$	$0.51507392x10^{-4}$
n C_4H_{10}	$0.57302344x10^4$	$0.75117737x10^2$	$-0.13120884x10^0$	$0.10517908x10^{-3}$
i C_5H_{12}	$0.83081953x10^4$	$0.75267792x10^2$	$-0.12945843x10^0$	$0.10845697x10^{-3}$
n C_5H_{12}	$0.12804211x10^5$	$0.61654007x10^2$	$-0.97365201x10^{-1}$	$0.84398722x10^{-4}$
n C_6H_{14}	$0.23001684x10^5$	$0.27744919x10^2$	$-0.31545494x10^{-1}$	$0.49981289x10^{-4}$
n C_7H_{16}	$0.14876816x10^5$	$0.59342438x10^2$	$-0.81853271x10^{-1}$	$0.81429855x10^{-4}$
n C_8H_{18}	$0.32793215x10^5$	$- 0.35040283x10^2$	$0.11162955x10^0$	$-0.42647429x10^{-4}$
n C_9H_{20}	$0.47024656x10^5$	$- 0.95395035x10^2$	$0.24547529x10^0$	$-0.13209638x10^{-3}$
n $C_{10}H_{22}$	$0.55238211x10^5$	$- 0.13195618x10^3$	$0.32518369x10^0$	$-0.18188384x10^{-3}$

$H_i = C_{1i} + C_{2i}\ T + C_{3i}\ T^2 + C_{4i}\ T^3$ (T in °R) Btu/lb mole.

APPENDIX D

```
11
-1.  0.  0.  0.  0. 1.1 1.2 1.2 1.3 1.4 1.3
 0. -1.  0.  0.  0. 1.2 1.1 1.  1.3 1.1 1.
 0.  0. -1.  0.  0.  .1  .2  .3  .4  .5  .5
 0.  0.  0. -1.  0.  .2  .3  .4  .5  .6  .7
 0.  0.  0.  0. -1.  .3  .4  .5  .6  .7  .8
 .1  .2  .3  .4  .5  .6  .7  .8  .9 1.  1.2
 .2  .3  .4  .5  .6  .7  .8  .9 1.  1.1 1.2
 .3  .4  .5  .6  .7  .8  .9 1.  1.1 1.2 1.3
 .4  .5  .6  .7  .8  .9 1.  1.1 1.2 1.3 1.4
 .5  .6  .7  .8  .9 1.  1.1 1.2 1.3 1.4 1.5
 .6  .7  .8  .9 1.  1.1 1.2 1.3 1.4 1.5 1.6
```

Appendix D (continued)

```
21
-1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.1 1.2 1.2 1.3 1.4 1.3 1.5 1.6 1.7 1.7 1.8
0.  -1.  0.  0.  0.  0.  0.  0.  0.  0.  1.2 1.1 1.  1.3 1.1 1.  1.8 1.2 1.9 2.1 1.8
0.   0. -1.  0.  0.  0.  0.  0.  0.  0.  .1  .2  .3  .4  .5  .6  .7  .8  .9  1.  1.1
0.   0.  0. -1.  0.  0.  0.  0.  0.  0.  .2  .3  .4  .5  .5  .7  .8  .9  1.  1.2 .13
0.   0.  0.  0. -1.  0.  0.  0.  0.  0.  .3  .4  .5  .6  .7  .8  .9  1.  1.1 1.2 1.3
0.   0.  0.  0.  0. -1.  0.  0.  0.  0.  .4  .5  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4
0.   0.  0.  0.  0.  0. -1.  0.  0.  0.  .5  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5
0.   0.  0.  0.  0.  0.  0. -1.  0.  0.  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6
0.   0.  0.  0.  0.  0.  0.  0. -1.  0.  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7
0.   0.  0.  0.  0.  0.  0.  0.  0. -1.  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8
.1  .2  .3  .4  .5  .6  .7  .8  .9  1.  1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2
.2  .3  .4  .5  .5  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2
.3  .4  .5  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3
.4  .5  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4
.5  .6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5
.6  .7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6
.7  .8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7
.8  .9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
.9  1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9
1.  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.
1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.  2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.  3.1
```

Appendix D (continued)

```
31
-1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  0. 1.1 1.2 1.2 1.3 1.4 1.3 1.5 1.5 1.7 1.7 1.8 1.9 1. .9 .8 .7
0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.2 1.1 1. 1.3 1.1 1. 1.8 1.2 1.9 2.1 1.8 1.7 1.6 1.5 1.4 1.3
0. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  .1 .2 .3 .4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.2 1.1 1.
0. 0. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  .2 .3 .4 .5 .6 .7 .8 .9 1. 1.2 .13 .14 1.8 1.7 1.6 1.5
0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. .4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.4 1.3 1.2
0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0.  .4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.5 1.4 1.3
0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0.  .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.6 1.5 1.4
0. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 0. 0.  .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.7 1.6 1.5
0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 0.  .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 1.8 1.7 1.6
0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 1.9 1.8 1.7
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0. .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2. 1.9 1.8
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.1 2. 1.9
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 0.  14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 2. 1. .9 .8
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 0. 15. 14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 2. 1. .9
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. -1. 15. 15. 14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 2. 1.
.1 .2 .3 .4 .5 .6 .7 .8 .9 1. 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 1. 1. .9 .8 .7 .6 .5
.2 .3 .4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.5 .24 .23 .22 .21 .2
.3 .4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.5 2.4 2.3 2.2 2. 2.
.4 .5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.7 2.6 2.5 2.4 2.3 2.

.5 .6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 2.8 2.7 2.6 2.5 2.4 2.

.6 .7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 4.5 2.6 2.7 2.8 2.9 3. 2.9 2.8 2.7 2.6 2.5 2.

.7 .8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3. 3.1 3. 2.9 2.8 2.7 2.6 2.

.8 .9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3. 3.1 3.2 3. 3. 2.9 2.8 2.7 2.

.9 1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3. 3.1 3.2 3.3 3.2 .3 .3 .2 .2 .1
1. 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 4.8 2.9 3. 3.1 3.2 3.3 3.4 3.3 3.2 3. 3.1 3.

1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3. 3.1 3.2 3.3 3.4 3.5 3.4 2.3 2.2 2.1

1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3. 3.1 3.2 3.3 3.4 3.5 3.6 3.4 2.3 2.2 2.1

1.3 1.4 1.5 1.6 1.7 1.8 1.9 2. 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8-2.9 3. 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.6 3.4 3.3 3.2

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 2. 1. .9 .8 .7 .6
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 3. 2. 1. .9
1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 15. 14. 13. 12. 11. 10. 9. 8. 7. 6. 5. 4. 3. 2. 1.
```

APPENDIX E

ELEMENTS OF MATRICES FOR NONIDEAL SYSTEM

For nonideal system only the following elements need to evaluate from different expressions. Following expressions are evaluated using UNIQUAK model. Because of the lack of time and for model independent programme, this feature was not incorporated.

$$\frac{\partial o_{i,n}}{\partial v_{j,n}} = \frac{\delta_{ij} V_n - v_{i,n}}{V_n^2} + \frac{l_{i,n}}{L_n} \frac{\gamma^L_{i,n}}{\left(\gamma^V_{i,n}\right)^2} \frac{\partial \gamma^V_{i,n}}{\partial v_{j,n}}$$

$$\frac{\partial o_{i,n}}{\partial l_{i,n}} = - \frac{1}{\gamma^V_{i,n}} \left[\frac{l_{i,n}}{L_n} \frac{\partial \gamma^L_{i,n}}{\partial l_{j,n}} + \gamma^L_{i,n} \left\{ \frac{\delta_{ij} L_n - l_{i,n}}{L_n^2} \right\} \right]$$

$$\frac{\partial o_{i,n}}{\partial T_n} = - \frac{l_{i,n}}{L_n} \frac{\gamma^V_{i,n} \frac{\partial \gamma^L_{i,n}}{\partial T_n} - \gamma^L_{i,n} \frac{\partial \gamma^V_{i,n}}{\partial T_n}}{\left(\gamma^V_{i,m}\right)^2}$$

$$\gamma^V_{i,n} = \mathrm{Exp} \left[1 - \frac{r_i V_n}{\sum_k v_{k,n} r_k} + \ln \left(\frac{r_i V_n}{\sum_j v_{j,n} r_j} \right) \right.$$

$$- \frac{q_i Z}{2} \left\{ 1 - \frac{r_i}{q_i} \frac{\sum_j v_{j,n} q_j}{\sum_j v_{j,n} r_j} + \ln\left(\frac{r_i}{q_i}\right) \right.$$

$$+ \ln \left(\frac{\sum_j v_{j,n} q_j}{\sum_j v_{j,n} r_j} \right) + q_i \left\{ 1 - \ln \sum \left(\frac{v_{j,n} q_j \Lambda_{ij}}{\sum_k V_{k,n} q_k} \right) \right.$$

$$- \sum_K \frac{v_{k,n} q_K \Lambda_{k,i}}{\sum_m v_{m,n} q_m \Lambda_{k,m}}$$

Appendix E continued.

Similarly for $\gamma^{L}_{i,n}$ where v is substituted by l

$$
\begin{aligned}
\frac{\partial \gamma^{V}_{i,n}}{\partial v_{j,n}} = \gamma^{V}_{i,n} \Bigg[& - \frac{r_i \sum_k v_{k,n} r_k - V_n r_j}{\left(\sum_k v_{k,n} r_k\right)^2} + \\
& \frac{\sum_k v_{k,n} r_k - r_j V_n}{r_i V_n \left(\sum_k v_{k,n} r_k\right)} - \frac{Z}{2} q_i \left\{ - \frac{r_i}{q_i} \frac{\left(\sum_k v_{k,n} r_k\right) q_j - \left(\sum_k v_{k,n} q_k\right) r}{\left(\sum_k v_{k,n} r_k\right)^2} \right. \\
& \left. + \frac{q_j \sum_k (v_{k,n} r_k) - r_j \sum_k w_{k,n} q_k}{\left(\sum_k v_{k,n} q_k\right)\left(\sum_k v_{k,n} r_k\right)} \right\} \\
& - q_i \left\{ \frac{\sum_k (w_{k,n} q_k) \sum_m m_j q_m \Lambda_{j,m} - q_j \sum q_m \Lambda_{m,i} w_{m,n}}{\sum_m (v_{m,n} q_m \Lambda_{i,m}) \sum_k (v_{k,n} q_k)} \right. \\
& - q_k \Lambda_{ki} \frac{\delta_{kj} \sum \left(v_{m,n} q_m \Lambda_{k,m}\right) - v_{k,n} q_j \Lambda_{k,j}}{\left(\sum_m (v_{m,n} q_m \Lambda_{k,m})\right)^2}
\end{aligned}
$$

$$
O_{i,n} = \frac{v_{i,n}}{V_n} - \frac{l_{i,n}}{L_n} \frac{\gamma^{L}_{i,n}}{\gamma^{V}_{i,n}}
$$

$\Lambda_{j,i} = \exp(- \frac{a_{ji}}{RT})$ where a_{ji} are UNIQUAK parameter.

Appendix E continued.

$$\frac{\partial \gamma^{v}_{i,n}}{\partial T_n} = \gamma^{V}_{i,n} \left[- \frac{q_i \sum'_j {}_j \frac{\partial \Lambda_{j,i}}{\partial T_n}}{\sum_j \theta_j \Lambda_{i,j}} - \sum_K \frac{\sum_k \theta_j \Lambda_{k,j} \quad \theta_k \frac{\partial \Lambda_{k,j}}{\partial T_n}}{} \right.$$

$$\left. \frac{- \theta_k \Lambda_{k,j} \sum_j \theta_{j'} \frac{\partial \Lambda_{k,j}}{\partial T_m}}{\left(\sum_j \theta_j \Lambda_{i,j} \right)^2} \right\}$$

$$= - \frac{\gamma^{v}_{i,n}}{RT_m^2} \quad \frac{q_i \sum_j \theta_j \Lambda_{j,i} \Big|_n}{\sum_j \theta_j \Lambda_{n,j}} -$$

$$\sum_K \frac{\left(\sum_k \theta_j \Lambda_{k,j}\right) \theta_k \Lambda_{k,i} \Big|_n - \theta_k \Lambda_{k,j} \sum_j \theta_j \Lambda_{k,i} \Big|_n}{\left(\sum \theta_j \Lambda_{i,j} \right)^2}$$

```
GENERAL PROGRAM FOR DESIGN AND SIMULATION OF MULTICOMPONENT
MULTISTAGE EQUILIBRIUM SEPARATION PROCESSES
============================================================
DEVELOPED AND PROGRAMMED BY L. FIENK
============================================================
EXTENDED NAPHTHALI SANDHOLM METHOD
SPARSITY AND SYMMETRY EXPLOITATION
===========================================================
LISTING OF SYMBOLS ARE AS FOLLOWS
===========================================================
AL(J):LIQ. RATE OF THE J TH STAGE
V(J):VAP RATE OF THE J TH STAGE
SV(J,I): VAP. RATE OF COMPONENT I AT THE J TH STAGE
SL(J,I): LIQ. RATE OF COMPONENT I AT THE J TH STAGE
SQ(J):HEAT INPUT AT THE J TH STAGE
SS(J):FRACTION OF VAP. STREAM TAKEN FROM THE J TH STAGE
SSS(J): FRACTION OF LIQ STREAM TAKEN FROM THE J TH STAGE
ETA(J):MURPHREE EFFICIENCY OF THE J TH STAGE
SH(I): LIQ PHASE ENTHALPY OF COMPONENT I AT ANY STAGE
H(I): VAP. PHASE ENTHALPY OF COMPONENT I AT J TH STAGE
HF(I): ENTHALPY OF FEED AT ANY STAGE FOR COMPONENT I
HH(I): ENTHALPY OF COMPONENT I NEXT BOTTOM STAGE OF THE STAGE
TB : TEMPERATURE OF THE VAPOR ENTERING N TH PLATE
TO : TEMPERATURE OF THE LIQUID ENTERING INTO FIRST PLATE
SLO(I) : LIQ. RATE OF COMPONENT I ENTERING AT FIRST PLATE
SVD(I): VAPOR FLOW RATE OF COMPONENT I ENTERING INTO N TH PLATE
VB    :   TOTAL VAPOR RATE ENERING INTO THE N TH PLATE
UNDER CONSIDERATION
SHH(I):ENTHALPY OF COMPONENT I NEXT UP STAGE OF THE STAGE
UNDER CONSIDERATION
AK(I): EQILIBRIUM CONSTANT VALUE FOR COMPONENT I AT ANY STAGE
N: NUMBER OF STAGES
C: TOTAL NUMBER OF COMPONENTS INVOLVED FOR SEPARATION
SF(J,I):FEED RATE OF COMPONENT I AT THE J TH STAGE
TF(J):FEED TEMPERATURE OF THE J TH STAGE
Q : ENTHALPY BALNCE NORMALISATION FACTOR

============================================================
SUBROUTINE ENL   :COMPUTES LIQUID PHASE ENTHALPY OF ALL
                  COMPONENTS FOR A GIVEN STAGE
SUBROUTINE ENV   :COMPUTES VAPOR PHASE ENTHALPY OF ALL
                  COMPONENTS AT A GIVEN STAGE
........... DIST  : COMPUTES THE EQUILIBRIUM CONSTANT FOR ALL
                   COMPONENTS AT GIVEN STAGE
........... DDIST : COMPUTES THE DERIVATIVE OF EQUILIBRIUM
                    CONSTANT K W.R.T. TEMPERATURE FOR A STAGE
.......... DENL : COMPUTES THE DERIVATIVE OF THE LIQUID
                   PHASE ENTHALPY OF ALL COMPONENTS FOR A
                   PARTICULAR STAGE
.......... DENV : COMPUTES THE DERIVATIVE OF THE VAPOR PHASE
                   ENTHALPY OF ALL COMPONENTS FOR A PARTICULAR
                  STAGE
...........: BI    : COMPUTES THE INVERSION OF B SUBMATRIX
...........: B1I   : COMPUTE INVERSION OF B1 MATRIX
                   NEW VECTORS
============================================================
INTEGER C,CT
COMMON C,CT,N
COMMON /AREA1/SLO,SVB,TB,TO,VB/AREA2/SS,SSS,TF,ETA,SQ
COMMON /A1/SA/A2/SC/A3/SE
DIMENSION B(29,29),C1(29,29),PB(14),PC(14),C2(8,29,29)
```

```
1,F(29),F1(8,29),A(29,29),SVB(14),SA(14,4),SE(14,4)
2,SL(8,14),SV(8,14),AL(8),V(8),SS(8),SSS(8),
3SF(8,14),SH(14),H(14),SQ(8),SHH(14),HF(14),TF(8),
4ETA(8),AK(14),T(8),HH(14),PA(14),SLO(14),
5VD(20),SC(14,4)
==================================================================.
OPEN (UNIT=21,FILE='S.DAT')
==================================================================.
INPUT TERMS MAINLY GUESS VECTOR AND DATA SPECIFICATION
===============================================================.
READ(41,*)N,CT,C
READ(41,*)((SL(J,I),I=1,C),J=1,N)
READ(41,*)((SV(J,I),I=1,C),J=1,N)
READ(41,*)(V(J),J=1,N)
READ(41,*)(AL(J),J=1,N)
READ(41,*)(T(J),J=1,N)
READ(41,*)(TF(J),J=1,N)
READ(41,*)(SQ(J),J=1,N)
READ(41,*)(ETA(J),J=1,N)
READ(41,*)(SS(J),J=1,N)
READ(41,*)(SSS(J),J=1,N)
READ(41,*)R,Q,EPS
READ(41,*)((SF(J,I),I=1,C),J=1,N)
READ(41,*)(SLO(I),I=1,C)
READ(41,*)(SVB(I),I=1,C)

READ(41,*)TB,TO,VB
READ(41,*)((SA(I,J),J=1,4),I=1,C)
READ(41,*)((SC(I,J),J=1,4),I=1,C)
READ(41,*)((SE(I,J),J=1,4),I=1,C)
===========================================================
END OF INPUT SPECIFICATION
==============================
AEF=O.
IF(IN.GE.1)GOTO 701
WRITE(44,2210)
FORMAT(4X,'INITIAL GUESS OF FLOW RATES AND TEMPERATURE ARE:')
WRITE(44,2211)
FORMAT(/,4X,'VAPOR PHASE COMPONENT FLOW RATES ARE')
WRITE(44,2212)
FORMAT(4X,36('-'))
WRITE(44,2213)
FORMAT(4X,'COMPONENT NO.',10X,'STAGE NOS. ARE')
WRITE(44,4415)
FORMAT(4X,13('-'),10X,14('-'),/)
WRITE(44,2214)
FORMAT(20X,'1',12X,'2',12X,'3',12X,'4',12X,'5',12X,'6',12X,
1'7',12X,'8')
WRITE(44,2241)
FORMAT(4X,116('-'))
WRITE(44,4451)
FORMAT(//)
WRITE(44,332)(I,(SV(J,I),J= 1,N),I=1,C)
FORMAT((4X,I3,7X,8(E10.4,3X)),/)
WRITE(44,1219)
FORMAT(//,4X,'LIQ. COMP. FLOW RATES ARE')
WRITE(44,2212)
WRITE(44,2213)
WRITE(44,4415)
WRITE(44,2214)
WRITE(44,2241)
WRITE(44,332)(I,(SL(J,I),J=1,N),I=1,C)
WRITE(44,4440)
FORMAT(//,'INITIAL VAPOR AND LIQUID RATE AND  TEMPERATURE GUESS
1 ARE ',//)
WRITE(44,4467)
WRITE(44,4468)
WRITE(44,7979)
WRITE(44,13)(J,V(J),AL(J),T(J),J=1,N)
WRITE(44,6100)
FORMAT(//,20X,'COMPONENT ',6X,'TOP LIQ.FEED',10X,'BOTTOM VAP.FEE
1D.')
WRITE(44,6101)
FORMAT(20X,10('-'),4X,18('-'),4X,18('-'))
WRITE(44,6102)(I,SLO(I),SVB(I),I=1,C)
FORMAT((24X,I3,7X,2(E18.8,4X))/)
FORMAT(27X,'IIT KANPUR INDIA ')
FORMAT(27X,'1984')
WRITE(5,8)
WRITE(5,14)
WRITE(5,15)
FORMAT(/,4X,'DEVELOPED AND PROGRAMMED BY L. FIENK')
===============================================================
#  COMPUTATION OF AUGMENTED ERROR FUNCTION AT THE GUESS VECTOR
#  =============================================================
#   #  # CONTRIBUTION DUE TO MASS BALANCE # # #
         ---------------------------------
================================================================
EVALUATION OF LO(I) & V(N+1,I) FOR TOTAL CONDENSER
& TOTAL REBOILER
==================================================
O=1./O
END OF CONDENSER & REBOILER PART
IF ENTERING LIQ. AT PLATE 1 IS NOT AT T(1) ADJUST THIS TL
 ACCORDINGLY
TL=T(1)
DO 1 J=1,N-1
P1=1.+SS(J)
P2=1.+SSS(J)
IF(J.NE.1) GOTO 78
DO 1111 I=1,C
F1(1,I)=-(SLO(I)+SV(J+1,I)+SF(J,I)-P1*SV(J,I)-P2*SL(J,I))
AEF=AEF+F1(1,I)*F1(1,I)
CONTINUE
GOTO 1
DO 105 I=1,C
F1(J,I)=-(SL(J-1,I)+SV(J+1,I)+SF(J,I)-P1*SV(J,I)-P2*SL(J,I))
AEF=AEF+F1(J,I)*F1(J,I)
CONTINUE
CONTINUE
DO 405 I=1,C
F1(N,I)=-(SL(N-1,I)+SVB(I)+SF(N,I)-P1*SV(N,I)-P2*SL(N,I))
AEF=AEF+F1(N,I)*F1(N,I)
CONTINUE
WRITE(44,18)AEF
FORMAT(//,4X,'AEF AFTER MASS BALANCE=',E11.4)
#=================================================
#       END OF EVALUATION OF MASS BALANCE CONTRIBUTION
#        ===============================================
#        START OF EVALUATION OF THE CONTRIBUTION DUE TO
#        ENTHALPY BALNACE
#        ================================================
========================================================
E=0.
DO 2 J=1,N-1
TT=T(J)
P1=1.+SS(J)
P2=1.+SSS(J)
TFF=TF(J)
CALL ENL(TT,SH)
IF(J.NE.1) GOTO 601
CALL ENV(TT,H)
CALL ENL(TO,SHH)
CALL ENL(TFF,HF)
T2=T(J+1)
CALL ENV(T2,HH)
IF(J.NE.1) GOTO 26
DO 23 I=1,C
E=E+SLO(I)*SHH(I)+SV(2,I)*HH(I)+SF(1,I)*HF(I)
1-P2*SL(1,I)*SH(I)-P2*SV(1,I)*H(I)
SHH(I)=SH(I)
H(I)=HH(I)
CONTINUE
F1(1,CT)=(E+SO(1))
AEF=AEF+F1(1,CT)*F1(1,CT)
F1(1,CT)=-Q*F1(1,CT)
FORMAT(4X,'J=',I3,4X,'AEFOF ENTHALPY =',E11.4)
GO TO 1112
E=0.
DO 172 I=1,C
E=E+SL(J-1,I)*SHH(I)+SV(J+1,I)*HH(I)+SF(J,I)*HF(I)
1-P2*SL(J,I)*SH(I)-P1*SV(J,I)*H(I)
SHH(I)=SH(I)
H(I)=HH(I)
CONTINUE
F1(J,CT)=-(E+SO(J))
AEF=AEF+F1(J,CT)*F1(J,CT)
F1(J,CT)=Q*F1(J,CT)
WRITE(5,22)J,AEF
CONTINUE
TT=T(N)
TFF=TF(N)
CALL ENL(TT,SH)
CALL ENL(TFF,HF)
CALL ENV(TB,HH)
E=0.
DO 272 I=1,C
E=E+SL(N-1,I)*SHH(I)+SVB(I)*HH(I)+SF(N,I)*HF(I)-P2*SL(N,I)*SH(I)
1-P1*SV(N,I)*H(I)
CONTINUE
F1(N,CT)=-(E+SO(N))
AEF=AEF+F1(N,CT)*F1(N,CT)
F1(N,CT)=Q*F1(N,CT)
WRITE(44,22)N,AEF
WRITE(5,22)N,AEF
FFF=AEF
===========================================================
#        TO COMPUTE THE LIQUID & VAPOR FLOW RATE FOR ALL STAGES
#        ======================================================.
================================================================.
#        CONTRIBUTION DUE TO EQUILIBRIUM CONDITION CONSIDERING
#        EFFICIENCY OF ALL STAGES
#        ======================================================.
```

```
==============================================================
DO 4 J=1,N-1
TT=T(J)
CALL DIST(TT,AK)
DO 4 I=1,C
F1(J,C+I)=-(ETA(J)*AK(I)*SL(J,I)/AL(J)-SV(J,I)/V(J)+(1-ETA(J))*
1SV(J+1,I)/V(J+1))
AEF=AEF+F1(J,C+I)*F1(J,C+I)
CONTINUE
TT=T(N)
CALL DIST(TT,AK)
DO 404 I=1,C
F1(N,C+I)=-(ETA(N)*AK(I)*SL(N,I)/AL(N)-SV(N,I)/V(N)+(1.-ETA(N))*
1SVB(I)/VB)
AEF=AEF+F1(N,C+I)*F1(N,C+I)
CONTINUE
==============================================================
#        END OF ERROR FUNCTION EVALUATION
==============================================================
IF(FFF-EPS) 5,5,6
==============================================================
FIND THE DIRECTION OF THE N TH PLATE
  TT=T(1)
P1=1.+SS(1)
P2=1.+SSS(1)
CALL DIST(TT,AK)
CALL ENL(TT,SH)
CALL ENV(TT,H)
CALL DDIST(TT,PA)
DO 36 I=1,C
B(C+I,2*C+1)=PA(I)
CONTINUE
CALL DENL(TT,PA)
DO 38 I=1,C
B(I,I)=PA(I)
CONTINUE
CALL DENV(TT,PA)
DO 40 I=1,C
B(I,I+C)=PA(I)
CONTINUE
P3=1./AL(1)
TT=0.
B(CT,CT)=0.
AAL=P3^2
VV=1./V(1)^2
DO 41 I=1,C
B(I+C,2*C+1)=ETA(1)*B(C+I,2*C+1)*P3*SL(1,I)
B(2*C+1,I)=-P2*SH(I)*0
B(2*C+1,C+I)=-P1*H(I)*0
B(2*C+1,2*C+1)=-SV(1,I)*B(I,C+I)+B(2*C+1,2*C+1)
TT=TT-SL(1,I)*B(I,I)
CONTINUE
DO 81 I=1,C
DO 81 IJ=1,C
IF(I.EQ.IJ) GOTO 82
B(I+C,IJ)=-ETA(1)*AK(I)*SL(1,I)*AAL
B(I+C,IJ+C)=SV(1,I)*VV
GOTO 81
B(C+I,IJ)=ETA(1)*AK(I)*(AL(1)-SL(1,I))*AAL
B(C+I,IJ+C)=-(V(1)-SV(1,I))*VV
CONTINUE
```

```
B(2*C+1,2*C+1)=B(2*C+1,2*C+1)*P1+TT*P2
B(CT,CT)=B(CT,CT)*Q
DO 42 I=1,C
B(I,I) =-P2
B(I,C+I)= -P1
CONTINUE
END OF COMPUTATION OF B1
WRITE(5,44)((B(I,J),J=1,2*C+1),I=1,2*C+1)
FORMAT(//,4X,'ELEMENTS OF B MATRIX',/(4X,9(E11.4,3X)))
CALL GBI(B)
WRITE(5,44)((B(I,J),J=1,2*C+1),I=1,2*C+1)
COMPUTATION OF C
TT=T(2)
CALL DENV(TT,PA)
CALL ENV(TT,HH)
P3=1/V(2)^2
C1(CT,CT)=0.
DO 112 I=1,C
C1(CT,C+I)=HH(I)*Q
C1(CT,CT)=C1(CT,CT)+SV(2,I)*PA(I)
C1(I,C+I)=1.
CONTINUE
DO 83 I=1,C
DO 83 II=1,C
IF(I.EQ.II)GOTO84
C1(I+C,II+C)=(1-ETA(1))*SV(2,I)*P3
GOTO 83
C1(I+C,II+C)=(1-ETA(1))*(V(2)-SV(2,I))*P3
CONTINUE
C1(CT,CT)=C1(CT,CT)*Q
WRITE(5,301)((C1(I,J),J=1,CT),I=1,CT)
FORMAT(//,2X,'ELEMENTS OF C MATRIX',/(4X,9(E11.4,3X)))
WRITE(5,44)((B(I,J),J=1,CT),I=1,CT)
CALL BCMUL(B,C1)
DO 111 I=1,CT
F(I)=F1(1,I)
DO 111 II=1,CT
C2(1,I,II)=C1(I,II)
CONTINUE
WRITE(5,301)((C1(I,J),J=1,CT),I=1,CT)
WRITE(5,305)(F(I),I=1,CT)
FORMAT(//,2X,'ELEMENETS OF ERROR VECTOR',/(4X,9(E13.6,3X)))
CALL MTMUL1(B,F)
DO 110 I=1,CT
F1(1,I)=F(I)
CONTINUE
WRITE(5,305)(F(I),I=1,CT)
START OF 2ND AND ONWORD STAGES

DO 54 J=2,N
P1=1.+SS(J)
P2=1.+SSS(J)
TT=T(J-1)
CALL DENL(TT,H)
AA=0.
CALL ENL(TT,SHH)
DO 55 I=1,C
AA=AA+SL(J-1,I)*H(I)
A(CT,I)=SHH(I)*Q
A(I,I)=1.
CONTINUE
```

```
A(CT,CT)=AA*Q
END OF COMPUTATION OF ELEMENTS OF A MATRIX
WRITE(5,307)((A(I,IJ),IJ=1,CT),I=1,CT)
FORMAT(//,4X,'ELEMENTS OF A MATRIX',/(4X,9(E11.4,3X)))
CALL ACMUL(A,C1)
WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
CALL AFMUL(A,F)
WRITE(5,305)(F(I),I=1,CT)
P3=1./AL(J)
TT=T(J)
DO 201 I=1,CT
DO 201 II=1,CT
B(I,II)=0.
CONTINUE
CALL DIST(TT,AK)
CALL ENL(TT,SH)
CALL ENV(TT,H)
CALL DDIST(TT,PA)
DO 60 I=1,C
B(C+I,CT)=PA(I)
CONTINUE
CALL DENV(TT,PA)
DO 61 I=1,C
B(I,C+I)=PA(I)
CONTINUE
CALL DENL(TT,PA)
DO 63 I=1,C
B(I,I)=PA(I)
CONTINUE
AAL=P3^2
VV=1/V(J)^2
TT=0.
B(CT,CT)=0.
DO 57 I=1,C
B(I+C,CT)=ETA(J)*B(C+I,CT)*P3*SL(J,I)
B(CT,I)=-P2*SH(I)*Q
B(CT,C+I)=-P1*H(I)*Q
B(CT,CT)=B(CT,CT)-SV(J,I)*B(I,C+I)
TT=TT-SL(J,I)*B(I,I)
CONTINUE
DO 85 I=1,C
DO 85 II=1,C
IF(I.EQ.II) GOTO 86
B(I+C,II)=-ETA(J)*AK(I)*SL(J,I)*AAL
B(I+C,II+C)=SV(J,I)*VV
GOTO 85
B(C+I,II)=ETA(J)*AK(I)*(AL(J)-SL(J,I))*AAL
B(C+I,II+C)=-(V(J)-SV(J,I))*VV
CONTINUE
B(CT,CT)=B(CT,CT)*P1+TT*P2
B(CT,CT)=B(CT,CT)*Q
DO 58 I=1,C
B(I,I)=-P2
B(I,C+I)=-P1
CONTINUE
WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
CALL SUBMAT(B,C1)
WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
CALL GBI(B)
WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
DO 160 I=1,CT
DO 160 II=1,CT
A(I,II)=0.
C1(I,II)=0.
CONTINUE
IF(J.EQ.N) GOTO 70
TT=T(J+1)
CALL ENV(TT,HH)
P3=1/V(J+1)**2
THESE A ELEMENTS ARE LEMENTS OF C1 SUBMATRIX
DO 56 I=1,C
C1(CT,C+I)=HH(I)*Q
CONTINUE
CALL DENV(TT,PA)
DO 59 I=1,C
A(CT,CT)=SV(J+1,I)*PA(I)+A(CT,CT)
C1(I,I+C)=1.
CONTINUE
C1(CT,CT)=Q*A(CT,CT)
DO 87 I=1,C
DO 87 II=1,C
IF(I.EQ.II) GOTO 88
C1(I+C,II+C)=(1.-ETA(J))*SV(J,I)*P3
C1(I+C,II+C)=(1.-ETA(J))*(V(J+1)-SV(J+1,I))*P3
CONTINUE
WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
CALL BCMUL(B,C1)
DO 67 III=1,CT
DO 67 II=1,CT
C2(J,III,II)=C1(III,II)
CONTINUE
WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
DO 69 I=1,CT
F(I)=F1(J,I)-F(I)
CONTINUE
WRITE(5,305)(F(I),I=1,CT)
TYPE*,((B(II,IJ),IJ=1,CT),II=1,CT)
FORMAT(4X,'F VALUES ARE:')
CALL MTMUL1(B,F)
WRITE(5,305)(F(I),I=1,CT)
DO 71 I=1,CT
F1(J,I)=F(I)
DO 71 II=1,CT
A(I,II)=0.
CONTINUE
CONTINUE
DO 72 JJ=1,N-1
J=N-JJ
DO 75 I=1,CT
F(I)=F1(J+1,I)
DO 75 II=1,CT
C1(I,II)=C2(J,I,II)
CONTINUE
WRITE(38,301)((C1(I,J),J=1,CT),I=1,CT)
WRITE(38,305)(F(I),I=1,CT)
CALL CFMUL(C1,F)
WRITE(5,305)(F(I),I=1,CT)
DO 76 I=1,CT
F1(J,I)=-F(I)+F1(J,I)
CONTINUE
CONTINUE
EFF=AEF
```

```
DF=-1.
DO 135 J=1,N
DO 136 I=1,C
SL(J,I)=SL(J,I)-DF*F1(J,I)
SV(J,I)=SV(J,I)-DF*F1(J,I+C)
CONTINUE
T(J)=T(J)-OF*F1(J,CT)
CONTINUE
IN=IN+1
WRITE(44,200)IN
FORMAT(/,'ITERATION NUMBER=',I3)
AEF=0.
DO 25 J=1,N
V(J)=0.
AL(J)=0.
CONTINUE
DO 202 I=1,CT
DO 202 II=1,CT
B(I,II)=0.
            C1(I,II)=0.
CONTINUE
DO 371 J=1,N
DO 371 I=1,C
V(J)=V(J)+SV(J,I)
AL(J)=AL(J)+SL(J,I)
CONTINUE
DO 444 J=1,N
CONTINUE
GO TO 16
WRITE(44,9)
FORMAT(/,'SUCESSFUL CONVERGENT')
WRITE(44,20)IN
FORMAT(/,'ITERATION NUMBER=',I3)
WRITE(44,2410)
FORMAT(4X,'FINAL VALUES  OF FLOW RATES AND TEMPERATURE ARE:')
WRITE(44,2411)
FORMAT(/,4X,'VAPOR PHASE COMPONENT FLOW RATES ARE')
WRITE(44,2412)
FORMAT(4X,36('-'))
WRITE(44,2413)
FORMAT(4X,'COMPONENT NO.',10X,'STAGE NOS. ARE')
WRITE(44,4315)
FORMAT(4X,13('-'),10X,14('-'),/)
WRITE(44,2414)
FORMAT(20X,'1',12X,'2',12X,'3',12X,'4',12X,'5',12X,'6',14X,
1'7',12X,'8')
WRITE(44,2441)
FORMAT(4X,116('-'))
WRITE(44,4351)
FORMAT(//)
WRITE(44,392)(I,(SV(J,I),J= 1,N),I=1,C)
FORMAT((4X,I3,7X,8(E10.4,3X)),/)
WRITE(44,1319)
FORMAT(//,4X,'LIQ. COMP. FLOW RATES ARE')
WRITE(44,2412)
WRITE(44,2413)
WRITE(44,4315)
WRITE(44,2414)
WRITE(44,2441)
WRITE(44,392)(I,(SL(J,I),J=1,N),I=1,C)
WRITE(44,4449)
```

```
9     FORMAT(//,4X,'VAPOR AND LIQUID RATE AND TEMPERATURE ARE ',//)
      WRITE(44,4467)
7     FORMAT(4X,42('-'))
      WRITE(44,4468)
8     FORMAT(4X,'STAGE NUMBER',4X,'     VAPOR RATE      ',4X,'   LIQUID
     1 RATE    ',4X,'          TEMPERATURE   ')
      WRITE(44,7979)
9     FORMAT(4X,12('-'),4X,19('-'),4X,17('-'),4X,19('-'),/)
      WRITE(44,13)(J,V(J),AL(J),T(J),J=1,N)
      FORMAT(8X,I3,5X,E19.8,4X,E17.8,4X,E19.8)
      IFAIL=1
      STOP
      END
      SUBROUTINE MTMUL2(A,B)
      INTEGER C,CT
      COMMON C,CT,N
      DIMENSION A(CT,CT),B(CT,CT),D(29,29)
      DO 1 I=1,CT
      DO 1 J=1,CT
      DO 1 IJ=1,CT
      D(I,J)=D(I,J)+A(I,IJ)*B(IJ,J)
      CONTINUE
      DO 2 I=1,CT
               DO 2 J=1,CT
      B(I,J)=D(I,J)
      D(I,J)=0.
      CONTINUE
      RETURN
      END
      SUBROUTINE MTMUL1(A,B)
      INTEGER C,CT
      COMMON C,CT,N
      DIMENSION A(CT,CT),B(CT),D(29)
      DO 1 I=1,CT
      DO 1 II=1,CT
      D(I)=A(I,II)*B(II)+D(I)
      CONTINUE
      DO 2 I=1,CT
      B(I)=D(I)
      D(I)=0.
      CONTINUE
      RETURN
      END
      SUBROUTINE ENL(T,EL)
      INTEGER C,CT
      COMMON C,CT,N
      COMMON /A2/SC
      DIMENSION EL(C),SC(14,4)
      DO 1 I=1,C
      EL(I)=SC(I,1)+SC(I,2)*T+SC(I,3)*T^2+SC(I,4)*T^3
      CONTINUE
      RETURN
      END
      SUBROUTINE DENL(T,EL)
      INTEGER C,CT
      COMMON C,CT,N
      COMMON /A2/SC
      DIMENSION EL(C),SC(14,4)
      DO 1 I=1,C
      EL(I)=SC(I,2)+2.*SC(I,3)*T+3.*SC(I,4)*T^2
      CONTINUE
```

```
RETURN
END
SUBROUTINE ENV(T,EV)
INTEGER C,CT
COMMON C,CT,N
COMMON /A3/SE
DIMENSION EV(C),SE(14,4)
DO 1 I=1,C
EV(I)=SE(I,1)+SE(I,2)*T+SE(I,3)*T^2+SE(I,4)*T^3
CONTINUE
RETURN
END
SUBROUTINE DENV(T,EV)
INTEGER C,CT
COMMON C,CT,N
COMMON /A3/SE
DIMENSION EV(C),SE(14,4)
DO 1 I=1,C
EV(I)=SE(I,2)+SE(I,3)*2.*T+3.*SE(I,4)*T^2
CONTINUE
RETURN
END
SUBROUTINE SUBMAT(B,C1)
INTEGER C,CT
COMMON C,CT,N
DIMENSION B(CT,CT),C1(CT,CT)
DO 1 I=1,CT
DO 1 J=1,CT
B(I,J)=B(I,J)-C1(I,J)
CONTINUE
RETURN
END
SUBROUTINE DIST(T,AK)
INTEGER C,CT
COMMON C,CT,N
COMMON /A1/SA
DIMENSION AK(C),SA(14,4)
DO 1 I=1,C
AK(I)=T*(SA(I,1)+SA(I,2)*T+SA(I,3)*T^2+SA(I,4)*T^3)^3
CONTINUE
RETURN
END
SUBROUTINE DDIST(T,AK)
INTEGER C,CT
COMMON C,CT,N
COMMON /A1/SA
DIMENSION AK(C),SA(14,4)
DO 1 I=1,C
AK(I)=(SA(I,1)+SA(I,2)*T+SA(I,3)*T^2+SA(I,4)*T^3)^3+3.*T*(
1SA(I,2)+2.*SA(I,3)*T+3.*SA(I,4)*T^2)*(SA(I,1)+SA(I,2)*T
2+SA(I,3)*T^2+SA(I,4)*T^3)^2
CONTINUE
RETURN
END
INVERSION OF GENERAL B MATRIX WITH SPARCITY EXPLOITATION
SUBROUTINE GBI(B)
INTEGER C,CT
COMMON C,CT,N
DIMENSION B(CT,CT),BS(14,14),BS1(14,14),PA(14),PB(14)
DO 1 I=1,C
PA(I)=1./B(I,I)
```

```
CONTINUE
DO 2 I=1,C
DO 2 J=1,C
BS(I,J)=PA(I)*B(I,J+C)
CONTINUE
DO 3 I=1,C
DO 3 J=1,C
DO 3 IJ=1,C
BS1(I,J)=BS1(I,J)+B(I+C,IJ)*BS(IJ,J)
CONTINUE
DO 4 I=1,C
DO 4 J=1,C
BS1(I,J)=B(I+C,J+C)-BS1(I,J)
CONTINUE
CALL FMI(BS1)
DO 5 I=1,C
DO 5 J=1,C
B(I+C,J+C)=BS1(I,J)
BS1(I,J)=B(I+C,J)*PA(J)
B(I,J+C)=0.
B(I+C,J)=0.
CONTINUE
DO 6 I=1,C
DO 6 J=1,C
B(I,J)=0.
DO 6 IJ=1,C
B(I,J+C)=B(I,J+C)-BS(I,IJ)*B(IJ+C,J+C)
CONTINUE
DO 7 I=1,C
DO 7 J=1,C
BS(I,J)=0.
DO 7 IJ=1,C
B(I+C,J)=B(I+C,J)-B(I+C,C+IJ)*BS1(IJ,J)
B(I,J)=B(I,J)-B(I,IJ+C)*BS1(IJ,J)
CONTINUE
DO 9 I=1,C
BS(I,I)=PA(I)
PA(I)=0.
DO 9 J=1,C
B(I,J)=B(I,J)+BS(I,J)
CONTINUE
DO 10 I=1,C
DO 10 II=1,C
PA(I)=PA(I)+B(I,II)*B(II,CT)+B(I,II+C)*B(II+C,CT)
PB(I)=PB(I)+B(I+C,II)*B(II,CT)+B(I+C,II+C)*B(II+C,CT)
CONTINUE
DO 11 I=1,C
 BB=BB+B(CT,I)*PA(I)+B(CT,I+C)*PB(I)
CONTINUE
B(CT,CT)=1./(B(CT,CT)-BB)
DO 12 I=1,C
B(I,CT)=-PA(I)*B(CT,CT)
B(I+C,CT)=-PB(I)*B(CT,CT)
PA(I)=0.
PB(I)=0.
CONTINUE
DO 13 J=1,C
DO 13 II=1,C
PA(J)=PA(J)+B(CT,II)*B(II,J)+B(CT,II+C)*B(II+C,J)
PB(J)=PB(J)+B(CT,II)*B(II,J+C)+B(CT,II+C)*B(II+C,J+C)
CONTINUE
```

```
DO 14 I=1,C
B(CT,I)=-B(CT,CT)*PA(I)
B(CT,I+C)=-B(CT,CT)*PB(I)
CONTINUE
DO 15 I=1,C
DO 15 J=1,C
B(I,J)=B(I,J)-B(I,CT)*PA(J)
B(I,J+C)=B(I,J+C)-B(I,CT)*PB(J)
B(I+C,J)=B(I+C,J)-B(I+C,CT)*PA(J)
B(I+C,J+C)=B(I+C,J+C)-B(I+C,CT)*PB(J)
BS1(I,J)=0.
CONTINUE
BB=0.
DO 22 I=1,C
PB(I)=0.
CONTINUE
RETURN
END
CORRECTED VERSION OF PRODUCT FORM OF INVERSE
A MATRIX IS TO BE INVERTED
INVERTED MATRIX IS TT AS WELL AS A
================================================
SUBROUTINE FMI(A)
INTEGER C,CT
COMMON C,CT,N
DIMENSION A(C,C),TT(14,14),UI(14,14),T(14,14),ETA(14,14)
DO 1 I=1,C
UI(I,I)=1.
DO 1 J=1,C
TT(I,J)=0.
CONTINUE
DO 5 K=1,C
ETA(K,K)=1./A(K,K)
DO 4 I=1,C
IF(I.EQ.K) GOTO 4
ETA(I,K)=-A(I,K)/A(K,K)
CONTINUE
DO 6 I=1,C
ETA(I,K)=ETA(I,K)-UI(I,K)
CONTINUE
DO 7 I=1,C
DO 7 J=1,C
T(I,J)=UI(I,J)+ETA(I,K)*UI(K,J)
CONTINUE
DO 8 I=1,C
DO 8 J=1,C
ETA(I,J)=0.
IF(K.NE.1) GOTO 8
TT(I,J)=T(I,J)
CONTINUE
DO 9 I=1,C
DO 9 J=K,C
DO 9 IJ=1,C
ETA(I,J)=T(I,IJ)*A(IJ,J)+ETA(I,J)
CONTINUE
DO 10 I=1,C
DO 10 J=1,C
A(I,J)=ETA(I,J)
ETA(I,J)=0.
CONTINUE
IF(K.EQ.1) GOTO 5
```

```
DO 11 I=1,C
DO 11 J=1,C
DO 11 IJ=1,C
ETA(I,J)=T(I,IJ)*TT(IJ,J)+ETA(I,J)
CONTINUE
DO 12 I=1,C
DO 12 J=1,C
TT(I,J)=ETA(I,J)
ETA(I,J)=0.
CONTINUE
CONTINUE
DO 14 I=1,C
DO 14 J=1,C
A(I,J)=TT(I,J)
CONTINUE
RETURN
END
INVERSION B1 MATRIX
DURING INVERSION SPARCITY IS EXPLOITED AND COMPUTATIONALLY
FASTER THAN AS USUAL INVERSION
SUBROUTINE B1I(B)
INTEGER C,CT
COMMON C,CT,N
DIMENSION B(CT,CT),SB(14,14),PA(14),PB(14)
DO 1 I=1,C
PA(I)=1./B(I,I)
PB(I)=PA(I)*B(I,I+C)
CONTINUE
DO 2 I=1,C
DO 2 J=1,C
SB(I,J)=B(I+C,J)*PB(J)
SB(I,J)=B(I+C,J+C)-SB(I,J)
CONTINUE
CALL FMI(SB)
DO 3 I=1,C
DO 3 J=1,C
B(I+C,J+C)=SB(I,J)
B(I,J+C)=-PB(I)*SB(I,J)
SB(I,J)=0.
CONTINUE
DO 4 I=1,C
DO 4 J=1,C
B(I+C,J)=-B(I+C,J)*PA(J)
B(I,J)=0.
CONTINUE
DO 16 I=1,C
B(I,I)=PA(I)
DO 16 J=1,C
DO 16 IJ=1,C
SB(I,J)=SB(I,J)+B(I,IJ+C)*B(IJ+C,J)
CONTINUE
DO 17 I=1,C
DO 17 J=1,C
B(I,J)=B(I,J)+SB(I,J)
SB(I,J)=0.
CONTINUE
DO 18 I=1,C
DO 18 J=1,C
DO 18 IJ=1,C
SB(I,J)=SB(I,J)+B(I+C,IJ+C)*B(IJ+C,J)
CONTINUE
```

```
DO 5 I=1,C
PA(I)=0.
PB(I)=0.
DO 5 J=1,C
B(I+C,J)=SB(I,J)
SB(I,J)=0.
CONTINUE
DO 10 I=1,C
DO 10 II=1,C
PA(I)=PA(I)+B(I,II)*B(II,CT)+B(I,II+C)*B(II+C,CT)
PB(I)=PB(I)+B(I+C,II)*B(II,CT)+B(I+C,II+C)*B(II+C,CT)
CONTINUE
DO 11 I=1,C
 BB=BB+B(CT,I)*PA(I)+B(CT,I+C)*PB(I)
CONTINUE
B(CT,CT)=1./(B(CT,CT)-BB)
DO 12 I=1,C
B(I,CT)=-PA(I)*B(CT,CT)
B(I+C,CT)=-PB(I)*B(CT,CT)
PA(I)=0.
PB(I)=0.
CONTINUE
DO 13 J=1,C
DO 13 II=1,C
PA(J)=PA(J)+B(CT,II)*B(II,J)+B(CT,II+C)*B(II+C,J)
PB(J)=PB(J)+B(CT,II)*B(II,J+C)+B(CT,II+C)*B(II+C,J+C)
CONTINUE
DO 14 I=1,C
B(CT,I)=-B(CT,CT)*PA(I)
B(CT,I+C)=-B(CT,CT)*PB(I)
CONTINUE
DO 15 I=1,C
DO 15 J=1,C
B(I,J)=B(I,J)-B(I,CT)*PA(J)
B(I,J+C)=B(I,J+C)-B(I,CT)*PB(J)
B(I+C,J)=B(I+C,J)-B(I+C,CT)*PA(J)
B(I+C,J+C)=B(I+C,J+C)-B(I+C,CT)*PB(J)
CONTINUE
BB=0.
RETURN
END
SUBROUTINE ACMUL(A,C1)
INTEGER C,CT
COMMON C,CT,N
DIMENSION A(CT,CT),C1(CT,CT)
DO 1 I=1,C
DO 1 J=1,C+1
C1(I+C,J+C)=0.
CONTINUE
DO 3 I=1,C
DO 2 J=1,C
C1(1,I)=C1(1,I)+A(CT,J)*C1(J,I+C)
CONTINUE
C1(CT,C+I)=C1(1,I)+A(CT,CT)*C1(CT,I+C)
C1(1,I)=0.
AA=AA+A(CT,I)*C1(I,CT)
CONTINUE
C1(CT,CT)=AA+A(CT,CT)*C1(CT,CT)
AA=0.
RETURN
END
```

```
SUBROUTINE AFMUL(A,F)
INTEGER C,CT
COMMON C,CT,N
DIMENSION A(CT,CT),F(CT)
DO 1 I=1,C
F(C+I)=0.
AA=AA+A(CT,I)*F(I)
CONTINUE
F(CT)=AA+F(CT)*A(CT,CT)
AA=0.
RETURN
END
SUBROUTINE BCMUL(B,C1)
INTEGER C,CT
COMMON C,CT,N
DIMENSION B(CT,CT),C1(CT,CT)
DO 2 I=1,CT
DO 2 J=1,C
DO 2 K=1,CT
C1(I,J)=C1(I,J)+B(I,K)*C1(K,J+C)
CONTINUE
DO 1 I=1,CT
DO 1 J=1,C
C1(I,J+C)=C1(I,J)
C1(I,J)=0.
CONTINUE
DO 4 I=1,CT
DO 4 K=1,CT
C1(I,1)=C1(I,1)+B(I,K)*C1(K,CT)
CONTINUE
DO 3 I=1,CT
C1(I,CT)=C1(I,1)
C1(I,1)=0.
CONTINUE
RETURN
END
SUBROUTINE CFMUL(C1,F)
INTEGER C,CT
COMMON C,CT,N
DIMENSION C1(CT,CT),F(CT)
DO 1 I=1,CT
DO 1 K=1,C+1
C1(I,1)=C1(I,1)+C1(I,K+C)*F(C+K)
CONTINUE
DO 2 I=1,CT
F(I)=C1(I,1)
C1(I,1)=0.
CONTINUE
RETURN
END
```

INPUT SPECIFICATION OF PROBLEM NUMBER 1

```
[illegible]
.15E-02 .35E-01 .27E-03 .96
.17E-02 .66E-01 .62E-03 .93
.16E-02 .77E-01 .1E-02 .91
.15E-02 .88E-01 .16E-02 .9
.15E-02 .94E-01 .22E-02 .9
.15E-02 .98E-01 .31E-02 .89
.15E-02 .1 .4E-02 .99
.15E-02 .1 .55E-02 .89
.15E-02 .1 .7E-02 .88
.15E-02 .1 .92E-02 .88
.15E-02 .1 .11E-01 .88
.15E-02 .1 .15E-01 .88
.15E-02 .1 .19E-01 .87
.14E-02 .99E-01 .23E-01 .87
.14E-02 .98E-01 .29E-01 .87
.14E-02 .96E-01 .37E-01 .86
.14E-02 .94E-01 .45E-01 .85
.14E-02 .91E-01 .56E-01 .85
.13E-02 .88E-01 .68E-01 .84
.138E-02 .85E-01 .82E-01 .83
.79E+02 .83E+02 .86E+02 .88E+02 .89E+02 .9E+02 .91E+02
.91E+02 .91E+02 .92E+02 .92E+02 .92E+02 .92E+02 .93E+02
.93E+02 .94E+02 .95E+02 .96E+02 .97E+02 .98E+02
.1E+03 .1E+03 .1E+03 .11E+03 .11E+03 .11E+03 .11E+03
.11E+03 .11E+03 .11E+03 .11E+03 .11E+03 .11E+03
.11E+03 .11E+03 .11E+03 .11E+03 .11E+03 .11E+03
.11E+03
145. 147. 150. 152. 155. 157. 160. 163. 166. 170. 173. 176.
179. 182. 185. 188. 191. 194. 197. 200.
125. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 200.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
5. 1. 1E-5
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. .0 .0 .0 .0
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
1. 1.8 3. 5.75
1.002 1.02 2.03 5.95
[illegible] .3 .4 1.5
0.13 .23 .43 1.9
50. 1.5 1.0 1E-6
55. 1.8 1. 1.5E-6
0. 0. 0. 100.
75. 15. 30. 0.
200. 125. 100.
```

```
GENERAL PROGRAM FOR DESIGN AND SIMULATION OF MULTICOMPONENT
MULTISTAGE EQUILIBRIUM SEPARATION PROCESSES
==============================================================
PROGRAMMED BY L. FIENK
==============================================================
NAPHTHALI SANDHOLM METHOD
=============================================================
LISTING OF SYMBOLS ARE AS FOLLOWS
=============================================================
AL(J):LIQ. RATE OF THE J TH STAGE
V(J):VAP RATE OF THE J TH STAGE
SV(J,I): VAP. RATE OF COMPONENT I AT THE J TH STAGE
SL(J,I): LIQ. RATE OF COMPONENT I AT THE J TH STAGE
SQ(J):HEAT INPUT AT THE J TH STAGE
SS(J):FRACTION OF VAP. STREAM TAKEN FROM THE J TH STAGE
SSS(J): FRACTION OF LIQ STREAM TAKEN FROM THE J TH STAGE
ETA(J):MURPHREE EFFICIENCY OF THE J TH STAGE
SH(I): LIQ PHASE ENTHALPY OF COMPONENT I AT ANY STAGE
H(I): VAP. PHASE ENTHALPY OF COMPONENT I AT J TH STAGE
HF(I): ENTHALPY OF FEED AT ANY STAGE FOR COMPONENT I
HH(I): ENTHALPY OF COMPONENT I NEXT BOTTOM STAGE OF THE STAGE
TB : TEMPERATURE OF THE VAPOR ENTERING N TH PLATE
TO : TEMPERATURE OF THE LIQUID ENTERING INTO FIRST PLATE
SLO(I) : LIQ. RATE OF COMPONENT I ENTERING AT FIRST PLATE
SVO(I): VAPOR FLOW RATE OF COMPONENT I ENTERING INTO N TH PLATE
VB    :  TOTAL VAPOR RATE ENERING INTO THE N TH PLATE
UNDER CONSIDERATION
SHH(I):ENTHALPY OF COMPONENT I NEXT UP STAGE OF THE STAGE
UNDER CONSIDERATION
AK(I): EQILIBRIUM CONSTANT VALUE FOR COMPONENT I AT ANY STAGE
N: NUMBER OF STAGES
C: TOTAL NUMBER OF COMPONENTS INVOLWED FOR SEPARATION
SF(J,I):FEED RATE OF COMPONENT I AT THE J TH STAGE
TF(J):FEED TEMPERATURE OF THE J TH STAGE
Q : ENTHALpY BALNCE NORMALISATION FACTOR

==============================================================
SUBROUTINE ENL   :COMPUTES LIQUID PHASE ENTHALPY OF ALL
                   COMPONENTS FOR A GIVEN STAGE
SUBROUTINE ENV   :COMPUTES VAPOR PHASE ENTHALPY OF ALL
                   COMPONENTS AT A GIVEN STAGE
.......... DIST   : COMPUTES THE EQUILIBRIUM CONSTANT FOR ALL
                    COMPONENTS AT GIVEN STAGE
.......... DDIST  : COMPUTES THE DERIVATIVE OF EQUILIBRIUM
                    CONSTANT K W.R.T. TEMPERATURE FOR A STAGE
.......... DENL : COMPUTES THE DERIVATIVE OF THE LIQUID
                   PHASE ENTHALPY OF ALL COMPONENTS FOR A
                   PARTICULAR STAGE
.......... DENV : COMPUTES THE DERIVATIVE OF THE VAPOR PHASE
                   ENTHALPY OF ALL COMPONENTS FOR A PARTICULAR
                  STAGE
........... BI    : COMPUTES THE INVERSION OF B SUBMATRIX
........... B1I   : COMPUTE INVERSION OF B1 MATRIX
                    NEW VECTORS
==============================================================
INTEGER C,CT
COMMON C,CT,N
COMMON /AREA1/SLO,SVB,TB,TO,VB/AREA2/SS,SSS,TF,ETA,SQ
COMMON /A1/SA/A2/SC/A3/SE
DIMENSION B(29,29),C1(29,29),PB(14),PC(14),C2(8,29,29)
```

```
      1,F(29),F1(8,29),A(29,29),SVB(14),SA(14,4),SE(14,4)
      2,SL(8,14),SV(8,14),AL(8),V(8),SS(8),SSS(8),
      3SF(8,14),SH(14),H(14),SQ(8),SHH(14),HF(14),TF(8),
      4ETA(8),AK(14),T(8),HH(14),PA(14),SLO(14),
      5VD(20),SC(14,4)
      ==============================================================
      OPEN (UNIT=21,FILE='S.DAT')
      ==============================================================
      INPUT TERMS MAINLY GUESS VECTOR AND DATA SPECIFICATION
      ============================================================
      READ(41,*)N,CT,C
      READ(41,*)((SL(J,I),I=1,C),J=1,N)
      READ(41,*)((SV(J,I),I=1,C),J=1,N)
      READ(41,*)(V(J),J=1,N)
      READ(41,*)(AL(J),J=1,N)
      READ(41,*)(T(J),J=1,N)
      READ(41,*)(TF(J),J=1,N)
      READ(41,*)(SQ(J),J=1,N)
      READ(41,*)(ETA(J),J=1,N)
      READ(41,*)(SS(J),J=1,N)
      READ(41,*)(SSS(J),J=1,N)
      READ(41,*)R,Q,EPS
      READ(41,*)((SF(J,I),I=1,C),J=1,N)
      READ(41,*)(SLO(I),I=1,C)
      READ(41,*)(SVB(I),I=1,C)

      READ(41,*)TB,TO,VB
      READ(41,*)((SA(I,J),J=1,4),I=1,C)
      READ(41,*)((SC(I,J),J=1,4),I=1,C)
      READ(41,*)((SE(I,J),J=1,4),I=1,C)
      =======================================================
      END OF INPUT SPECIFICATION
      =========================
      AEF=0.
      IF(IN.GE.1)GOTO 701
      WRITE(44,2210)
10    FORMAT(4X,'INITIAL GUESS OF FLOW RATES AND TEMPERATURE ARE:')
      WRITE(44,2211)
11    FORMAT(/,4X,'VAPOR PHASE COMPONENT FLOW RATES ARE')
      WRITE(44,2212)
12    FORMAT(4X,36('-'))
      WRITE(44,2213)
3     FORMAT(4X,'COMPONENT NO.',10X,'STAGE NOS. ARE')
      WRITE(44,4415)
5     FORMAT(4X,13('-'),10X,14('-'),/)
      WRITE(44,2214)
4     FORMAT(20X,'1',12X,'2',12X,'3',12X,'4',12X,'5',12X,'6',12X,
      1'7',12X,'8')
      WRITE(44,2241)
1     FORMAT(4X,116('-'))
      WRITE(44,4451)
1     FORMAT(//)
      WRITE(44,332)(I,(SV(J,I),J= 1,N),I=1,C)
      FORMAT((4X,I3,7X,8(E10.4,3X)),/)
      WRITE(44,1219)
9     FORMAT(//,4X,'LIQ. COMP. FLOW RATES ARE')
      WRITE(44,2212)
      WRITE(44,2213)
      WRITE(44,4415)
      WRITE(44,2214)
      WRITE(44,2241)
      WRITE(44,332)(I,(SL(J,I),J=1,N),I=1,C)
      WRITE(44,4440)
```

```
40     FORMAT(//,'INITIAL VAPOR AND LIQUID RATE AND  TEMPERATURE GUESS
       1 ARE ',//)
       WRITE(44,4467)
       WRITE(44,4468)
       WRITE(44,7979)
       WRITE(44,13)(J,V(J),AL(J),T(J),J=1,N)
       WRITE(44,6100)
00     FORMAT(//,20X,'COMPONENT ',6X,'TOP LIQ.FEED',10X,'BOTTOM VAP.FEE
       1D,')
       WRITE(44,6101)
01     FORMAT(20X,10('-'),4X,18('-'),4X,18('-'))
       WRITE(44,6102)(I,SLO(I),SVB(I),I=1,C)
02     FORMAT((24X,I3,7X,2(E18.8,4X))/)
       FORMAT(27X,'IIT KANPUR INDIA ')
       FORMAT(27X,'1984')
       WRITE(5,8)
       WRITE(5,14)
       WRITE(5,15)
       FORMAT(/,4X,'DEVELOPED AND PROGRAMMED BY L. FIENK')
       ================================================================
       #  COMPUTATION OF AUGMENTED ERROR FUNCTION AT THE GUESS VECTOR
       #  ==============================================================
       #  #  # CONTRIBUTION DUE TO MASS BALANCE # # #
                 ------------------------------------
       ================================================================
       EVALUATION OF LO(I) & V(N+1,I) FOR TOTAL CONDENSER
       & TOTAL REBOILER
       ================================================
       Q=1.
       Q=1./Q
       END OF CONDENSER & REBOILER PART
       IF ENTERING LIQ. AT PLATE 1 IS NOT AT T(1) ADJUST THIS TL
        ACCORDINGLY
       TL=T(1)
       T(N+1)=T(N)
       DO 1 J=1,N-1
       P1=1.+SS(J)
       P2=1.+SSS(J)
       IF(J.NE.1) GOTO 78
       DO 1111 I=1,C
       F1(1,I)=-(SLO(I)+SV(J+1,I)+SF(J,I)-P1*SV(J,I)-P2*SL(J,I))
       AEF=AEF+F1(1,I)*F1(1,I)
       FORMAT(4X,'AEF AFTER MASS BALANCE=',E11.4)
1      CONTINUE
       GOTO 1
       DO 105 I=1,C
       F1(J,I)=-(SL(J-1,I)+SV(J+1,I)+SF(J,I)-P1*SV(J,I)-P2*SL(J,I))
       AEF=AEF+F1(J,I)*F1(J,I)
       CONTINUE
       CONTINUE
       DO 405 I=1,C
       F1(N,I)=-(SL(N-1,I)+SVB(I)+SF(N,I)-P1*SV(N,I)-P2*SL(N,I))
       AEF=AEF+F1(N,I)*F1(N,I)
       CONTINUE
       WRITE(44,18)AEF
       #=============================================
       #      END OF EVALUATION OF MASS BALANCE CONTRIBUTION
       #        ==========================================
       #        START OF EVALUATION OF THE CONTRIBUTION DUE TO
       #        ENTHALPY BALNACE
       #        ==========================================
      ================================================================.
      E=0.
      DO 2  J=1,N-1
      TT=T(J)
      P1=1.+SS(J)
      P2=1.+SSS(J)
      TFF=TF(J)
      CALL  ENL(TT,SH)
      IF(J.NE.1) GOTO 601
      CALL  ENV(TT,H)
      CALL  ENL(TO,SHH)
1     CALL  ENL(TFF,HF)
      T2=T(J+1)
      CALL  ENV(T2,HH)
      IF(J.NE.1) GOTO 26
      DO 23  I=1,C
      E=E+SLO(I)*SHH(I)+SV(2,I)*HH(I)+SF(1,I)*HF(I)
      1-P2*SL(1,I)*SH(I)-P2*SV(1,I)*H(I)
      SHH(I)=SH(I)
      H(I)=HH(I)
      CONTINUE
      F1(1,CT)=(E+SO(1))
      AEF=AEF+F1(1,CT)*F1(1,CT)
      F1(1,CT)=-Q*F1(1,CT)
      FORMAT(4X,'J=',I3,4X,'AEFOF ENTHALPY =',E11.4)
      GO TO 1112
      E=0.
      DO 172  I=1,C
      E=E+SL(J-1,I)*SHH(I)+SV(J+1,I)*HH(I)+SF(J,I)*HF(I)
      1-P2*SL(J,I)*SH(I)-P1*SV(J,I)*H(I)
      SHH(I)=SH(I)
      H(I)=HH(I)
2     CONTINUE
      F1(J,CT)=-(E+SO(J))
      AEF=AEF+F1(J,CT)*F1(J,CT)
      F1(J,CT)=Q*F1(J,CT)
12    WRITE(5,22)J,AEF
      CONTINUE
      TT=T(N)
      TFF=TF(N)
      CALL  ENL(TT,SH)
      CALL  ENL(TFF,HF)
      CALL  ENV(TB,HH)
      E=0.
      DO 272  I=1,C
      E=E+SL(N-1,I)*SHH(I)+SVB(I)*HH(I)+SF(N,I)*HF(I)-P2*SL(N,I)*SH(I)
      1-P1*SV(N,I)*H(I)
      CONTINUE
      F1(N,CT)=-(E+SO(N))
      AEF=AEF+F1(N,CT)*F1(N,CT)
      F1(N,CT)=Q*F1(N,CT)
      FFF=AEF
      WRITE(44,22)N,AEF
      WRITE(5,22)N,AEF
      =======================================================.
      #        TO COMPUTE THE LIQUID & VAPOR FLOW RATE FOR ALL STAGES
      #        ==============================================.
      ================================================================.
      #        CONTRIBUTION DUE TO EQUILIBRIUM CONDITION CONSIDERING
      #        EFFICIENCY OF ALL STAGES
      #        ==============================================.
      ================================================================.
```

```
DO 4 J=1,N-1
TT=T(J)
CALL DIST(TT,AK)
DO 4 I=1,C
F1(J,C+I)=-(ETA(J)*AK(I)*SL(J,I)/AL(J)-SV(J,I)/V(J)+(1-ETA(J))*
1SV(J+1,I)/V(J+1))
AEF=AEF+F1(J,C+I)*F1(J,C+I)
CONTINUE
TT=T(N)
CALL DIST(TT,AK)
DO 404 I=1,C
F1(N,C+I)=-(ETA(N)*AK(I)*SL(N,I)/AL(N)-SV(N,I)/V(N)+(1.-ETA(N))*
1SVB(I)/VB)
AEF=AEF+F1(N,C+I)*F1(N,C+I)
CONTINUE
=================================================================.
#        END OF ERROR FUNCTION EVALUATION
=================================================================.
IF(FFF-EPS) 5,5,6
=================================================================.
FIND THE DIRECTION OF THE N TH PLATE
  TT=T(1)
P1=1.+SS(1)
P2=1.+SSS(1)
CALL DIST(TT,AK)
CALL ENL(TT,SH)
CALL ENV(TT,H)
CALL DDIST(TT,PA)
DO 36 I=1,C
B(C+I,2*C+1)=PA(I)
CONTINUE
CALL DENL(TT,PA)
DO 38 I=1,C
B(I,I)=PA(I)
CONTINUE
CALL DENV(TT,PA)
DO 40 I=1,C
B(I,I+C)=PA(I)
CONTINUE
P3=1./AL(1)
TT=0.
B(CT,CT)=0.
AAL=P3^2
VV=1./V(1)^2
DO 41 I=1,C
B(I+C,2*C+1)=ETA(1)*B(C+I,2*C+1)*P3*SL(1,I)
B(2*C+1,I)=-P2*SH(I)*Q
B(2*C+1,C+I)=-P1*H(I)*Q
B(2*C+1,2*C+1)=-SV(1,I)*B(I,C+I)+B(2*C+1,2*C+1)
TT=TT-SL(1,I)*B(I,I)
CONTINUE
DO 81 I=1,C
DO 81 IJ=1,C
IF(I.EQ.IJ) GOTO 82
B(I+C,IJ)=-ETA(1)*AK(I)*SL(1,I)*AAL
B(I+C,IJ+C)=SV(1,I)*VV
GOTO 81
B(C+I,IJ)=ETA(1)*AK(I)*(AL(1)-SL(1,I))*AAL
B(C+I,IJ+C)=-(V(1)-SV(1,I))*VV
CONTINUE
B(2*C+1,2*C+1)=B(2*C+1,2*C+1)*P1+TT*P2
```

```
      B(CT,CT)=B(CT,CT)*Q
      DO 42 I=1,C
      B(I,I) =-P2
      B(I,C+I)= -P1
2     CONTINUE
      END OF COMPUTATION OF B1
      WRITE(5,44)((B(I,J),J=1,2*C+1),I=1,2*C+1)
4     FORMAT(//,4X,'ELEMENTS OF B MATRIX',/(4X,9(E11.4,3X)))
      CALL GBI(B)
      WRITE(5,44)((B(I,J),J=1,2*C+1),I=1,2*C+1)
      COMPUTATION OF C
      TT=T(2)
      CALL DENV(TT,PA)
      CALL ENV(TT,HH)
      P3=1/V(2)*2
      C1(CT,CT)=0.
      DO 112 I=1,C
      C1(CT,C+I)=HH(I)*Q
      C1(CT,CT)=C1(CT,CT)+SV(2,I)*PA(I)
      C1(I,C+I)=1.
.2    CONTINUE
      DO 83 I=1,C
      DO 83 II=1,C
      IF(I.EQ.II)GOTO84
      C1(I+C,II+C)=(1-ETA(1))*SV(2,I)*P3
      GOTO 83
4     C1(I+C,II+C)=(1-ETA(1))*(V(2)-SV(2,I))*P3
3     CONTINUE
      C1(CT,CT)=C1(CT,CT)*Q
      WRITE(5,301)((C1(I,J),J=1,CT),I=1,CT)
01    FORMAT(//,2X,'ELEMENTS OF C MATRIX',/(4X,9(E11.4,3X)))
      WRITE(5,44)((B(I,J),J=1,CT),I=1,CT)
      CALL MTMUL2(B,C1)
      DO 111 I=1,CT
      F(I)=F1(1,I)
      DO 111 II=1,CT
      C2(1,I,II)=C1(I,II)
1     CONTINUE
      WRITE(5,301)((C1(I,J),J=1,CT),I=1,CT)
      WRITE(5,305)(F(I),I=1,CT)
5     FORMAT(//,2X,'ELEMENETS OF ERROR VECTOR',/(4X,9(E13.6,3X)))
      CALL MTMUL1(B,F)
      DO 110 I=1,CT
      F1(1,I)=F(I)
0     CONTINUE
      WRITE(5,305)(F(I),I=1,CT)
      START OF 2ND AND ONWORD STAGES

      DO 54 J=2,N
      P1=1.+SS(J)
      P2=1.+SSS(J)
      TT=T(J-1)
      CALL DENL(TT,H)
      AA=0.
      CALL ENL(TT,SHH)
      DO 55 I=1,C
      AA=AA+SL(J-1,I)*H(I)
      A(CT,I)=SHH(I)*Q
      A(I,I)=1.
      CONTINUE
      A(CT,CT)=AA*Q
      END OF COMPUTATION OF ELEMENTS OF A MATRIX
      WRITE(5,307)((A(I,IJ),IJ=1,CT),I=1,CT)
07    FORMAT(//,4X,'ELEMENTS OF A MATRIX',/(4X,9(E11.4,3X)))
      CALL MTMUL2(A,C1)
      WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
      CALL MTMUL1(A,F)
      WRITE(5,305)(F(I),I=1,CT)
      P3=1./AL(J)
      TT=T(J)
      DO 201 I=1,CT
      DO 201 II=1,CT
      B(I,II)=0.
01    CONTINUE
      CALL DIST(TT,AK)
      CALL ENL(TT,SH)
      CALL ENV(TT,H)
      CALL DDIST(TT,PA)
      DO 60 I=1,C
      B(C+I,CT)=PA(I)
0     CONTINUE
      CALL DENV(TT,PA)
      DO 61 I=1,C
      B(I,C+I)=PA(I)
1     CONTINUE
      CALL DENL(TT,PA)
      DO 63 I=1,C
      B(I,I)=PA(I)
3     CONTINUE
      AAL=P3^2
      VV=1/V(J)^2
      TT=0.
      B(CT,CT)=0.
      DO 57 I=1,C
      B(I+C,CT)=ETA(J)*B(C+I,CT)*P3*SL(J,I)
      B(CT,I)=-P2*SH(I)*Q
      B(CT,C+I)=-P1*H(I)*Q
      B(CT,CT)=B(CT,CT)-SV(J,I)*B(I,C+I)
      TT=TT-SL(J,I)*B(I,I)
7     CONTINUE
      DO 85 I=1,C
      DO 85 II=1,C
      IF(I.EQ.II) GOTO 86
      B(I+C,II)=-ETA(J)*AK(I)*SL(J,I)*AAL
      B(I+C,II+C)=SV(J,I)*VV
      GOTO 85
86    B(C+I,II)=ETA(J)*AK(I)*(AL(J)-SL(J,I))*AAL
      B(C+I,II+C)=-(V(J)-SV(J,I))*VV
85    CONTINUE
      B(CT,CT)=B(CT,CT)*P1+TT*P2
      B(CT,CT)=B(CT,CT)*Q
      DO 58 I=1,C
      B(I,I)=-P2
      B(I,C+I)=-P1
58    CONTINUE
      WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
      CALL SUBMAT(B,C1)
      WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
      CALL GBI(B)
      WRITE(5,44)((B(I,JI),JI=1,CT),I=1,CT)
      DO 160 I=1,CT
      DO 160 II=1,CT
      A(I,II)=0.
      C1(I,II)=0.
60    CONTINUE
      IF(J.EQ.N) GOTO 70
      TT=T(J+1)
      CALL ENV(TT,HH)
      P3=1/V(J+1)**2
      THESE A ELEMENTS ARE LEMENTS OF C1 SUBMATRIX
      DO 56 I=1,C
      C1(CT,C+I)=HH(I)*Q
6     CONTINUE
      CALL DENV(TT,PA)
      DO 59 I=1,C
      A(CT,CT)=SV(J+1,I)*PA(I)+A(CT,CT)
      C1(I,I+C)=1.
9     CONTINUE
      C1(CT,CT)=Q*A(CT,CT)
      DO 87 I=1,C
      DO 87 II=1,C
      IF(I.EQ.II) GOTO 88
      C1(I+C,II+C)=(1.-ETA(J))*SV(J,I)*P3
8     C1(I+C,II+C)=(1.-ETA(J))*(V(J+1)-SV(J+1,I))*P3
7     CONTINUE
      WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
      CALL MTMUL2(B,C1)
      DO 67 III=1,CT
      DO 67 II=1,CT
      C2(J,III,II)=C1(III,II)
7     CONTINUE
      WRITE(5,301)((C1(I,JI),JI=1,CT),I=1,CT)
9     DO 69 I=1,CT
      F(I)=F1(J,I)-F(I)
9     CONTINUE
      WRITE(5,305)(F(I),I=1,CT)
      TYPE*,((B(II,IJ),IJ=1,CT),II=1,CT)
      CALL MTMUL1(B,F)
      WRITE(5,305)(F(I),I=1,CT)
      DO 71 I=1,CT
      F1(J,I)=F(I)
      DO 71 II=1,CT
      A(I,II)=0.
1     CONTINUE
4     CONTINUE
```

```
      DO 72 JJ=1,N-1
      J=N-JJ
      DO 75 I=1,CT
      F(I)=F1(J+1,I)
      DO 75 II=1,CT
      C1(I,II)=C2(J,I,II)
5     CONTINUE
      WRITE(38,305)(F(I),I=1,CT)
      CALL MTMUL1(C1,F)
      WRITE(5,305)(F(I),I=1,CT)
      DO 76 I=1,CT
      F1(J,I)=-F(I)+F1(J,I)
6     CONTINUE
2     CONTINUE
      EFF=AEF
      OF=-1.
      DO 135 J=1,N
      DO 136 I=1,C
      SL(J,I)=SL(J,I)-OF*F1(J,I)
      SV(J,I)=SV(J,I)-OF*F1(J,I+C)
36    CONTINUE
      T(J)=T(J)-OF*F1(J,CT)
35    CONTINUE
      IN=IN+1
      WRITE(44,200)IN
0     FORMAT(/,'ITERATION NUMBER=',I3)
      AEF=0.
      DO 25 J=1,N
      V(J)=0.
      AL(J)=0.
      CONTINUE
      DO 202 I=1,CT
      DO 202 II=1,CT
      B(I,II)=0.
             C1(I,II)=0.
2     CONTINUE
      DO 371 J=1,N
      DO 371 I=1,C
      V(J)=V(J)+SV(J,I)
      AL(J)=AL(J)+SL(J,I)
1     CONTINUE
      DO 444 J=1,N
4     CONTINUE
      GO TO 16
      WRITE(44,9)
      FORMAT(/,'SUCESSFUL CONVERGENT')
      WRITE(44,20)IN
      FORMAT(/,'ITERATION NUMBER=',I3)
      WRITE(44,2410)
10    FORMAT(4X,'FINAL VALUES  OF FLOW RATES AND TEMPERATURE ARE:')
      WRITE(44,2411)
11    FORMAT(/,4X,'VAPOR PHASE COMPONENT FLOW RATES ARE')
      WRITE(44,2412)
12    FORMAT(4X,36('-'))
      WRITE(44,2413)
13    FORMAT(4X,'COMPONENT NO.',10X,'STAGE NOS. ARE')
      WRITE(44,4315)
15    FORMAT(4X,13('-'),10X,14('-'),/)
      WRITE(44,2414)
14    FORMAT(20X,'1',12X,'2',12X,'3',12X,'4',12X,'5',12X,'6',12X,
     1'7',12X,'8')
```

```
      WRITE(44,2441)
441   FORMAT(4X,116('-'))
      WRITE(44,4351)
351   FORMAT(//)
      WRITE(44,392)(I,(SV(J,I),J= 1,N),I=1,C)
92    FORMAT((4X,I3,7X,8(E10.4,3X)),/)
      WRITE(44,1319)
319   FORMAT(//,4X,'LIQ. COMP. FLOW RATES ARE')
      WRITE(44,2412)
      WRITE(44,2413)
      WRITE(44,4315)
      WRITE(44,2414)
      WRITE(44,2441)
      WRITE(44,392)(I,(SL(J,I),J=1,N),I=1,C)
      WRITE(44,4449)
449   FORMAT(//,4X,'VAPOR AND LIQUID RATE AND TEMPERATURE ARE ',//)
      WRITE(44,4467)
467   FORMAT(4X,42('-'))
      WRITE(44,4468)
468   FORMAT(4X,'STAGE NUMBER',4X,'     VAPOR RATE      ',4X,'   LIQUID
     1 RATE     ',4X,'        TEMPERATURE    ')
      WRITE(44,7979)
979   FORMAT(4X,12('-'),4X,19('-'),4X,17('-'),4X,19('-'),/)
      WRITE(44,13)(J,V(J),AL(J),T(J),J=1,N)
2     FORMAT(8X,I3,5X,E19.8,4X,E17.8,4X,E19.8)
      IFAIL=1
      STOP
      END
      CORRECTED VERSION OF PRODUCT FORM OF INVERSE
      A MATRIX IS TO BE INVERTED
      INVERTED MATRIX IS TT
      ================================================
      SUBROUTINE GBI(A)
      INTEGER C,CT
      COMMON C,CT,N
      DIMENSION A(CT,CT),TT(29,29),UI(29,29),T(29,29),ETA(29,29)
      DO 1 I=1,CT
      UI(I,I)=1.
      CONTINUE
      DO 5 K=1,CT
      ETA(K,K)=1./A(K,K)
      DO 4 I=1,CT
      IF(I.EQ.K) GOTO 4
      ETA(I,K)=-A(I,K)/A(K,K)
      CONTINUE
      DO 6 I=1,CT
      ETA(I,K)=ETA(I,K)-UI(I,K)
      CONTINUE
      DO 7 I=1,CT
      DO 7 J=1,CT
      T(I,J)=UI(I,J)+ETA(I,K)*UI(K,J)
      CONTINUE
      DO 8 I=1,CT
      DO 8 J=1,CT
      ETA(I,J)=0.
      IF(K.NE.1) GOTO 8
      TT(I,J)=T(I,J)
      CONTINUE
      DO 9 I=1,CT
      DO 9 J=K,CT
      DO 9 IJ=1,CT
```

```
ETA(I,J)=T(I,IJ)*A(IJ,J)+ETA(I,J)
CONTINUE
DO 10 I=1,CT
DO 10 J=1,CT
A(I,J)=ETA(I,J)
ETA(I,J)=0.
CONTINUE
IF(K.EQ.1) GOTO 5
DO 11 I=1,CT
DO 11 J=1,CT
DO 11 IJ=1,CT
ETA(I,J)=T(I,IJ)*TT(IJ,J)+ETA(I,J)
CONTINUE
DO 12 I=1,CT
DO 12 J=1,CT
TT(I,J)=ETA(I,J)
ETA(I,J)=0.
CONTINUE
CONTINUE
DO 21 I=1,CT
DO 21 J=1,CT
A(I,J)=TT(I,J)
CONTINUE
RETURN
END
SUBROUTINE MTMUL2(A,B)
INTEGER C,CT
COMMON C,CT,N
DIMENSION A(CT,CT),B(CT,CT),D(29,29)
DO 1 I=1,CT
DO 1 J=1,CT
DO 1 IJ=1,CT
D(I,J)=D(I,J)+A(I,IJ)*B(IJ,J)
CONTINUE
DO 2 I=1,CT
         DO 2 J=1,CT
B(I,J)=D(I,J)
D(I,J)=0.
CONTINUE
RETURN
END
SUBROUTINE MTMUL1(A,B)
INTEGER C,CT
COMMON C,CT,N
DIMENSION A(CT,CT),B(CT),D(29)
DO 1 I=1,CT
DO 1 II=1,CT
D(I)=A(I,II)*B(II)+D(I)
CONTINUE
DO 2 I=1,CT
B(I)=D(I)
D(I)=0.
CONTINUE
RETURN
END
SUBROUTINE SUBMAT(B,C1)
INTEGER C,CT
COMMON C,CT,N
DIMENSION B(CT,CT),C1(CT,CT)
DO 1 I=1,CT
DO 1 J=1,CT
```

```
B(I,J)=B(I,J)-C1(I,J)
CONTINUE
RETURN
END
SUBROUTINE ENL(T,EL)
INTEGER C,CT
COMMON C,CT,N
COMMON /A2/SC
DIMENSION EL(C),SC(14,4)
DO 1 I=1,C
EL(I)=SC(I,1)+SC(I,2)*T+SC(I,3)*T^2+SC(I,4)*T^3
CONTINUE
RETURN
END
SUBROUTINE DENL(T,EL)
INTEGER C,CT
COMMON C,CT,N
COMMON /A2/SC
DIMENSION EL(C),SC(14,4)
DO 1 I=1,C
EL(I)=SC(I,2)+2.*SC(I,3)*T+3.*SC(I,4)*T^2
CONTINUE
RETURN
END
SUBROUTINE ENV(T,EV)
INTEGER C,CT
COMMON C,CT,N
COMMON /A3/SE
DIMENSION EV(C),SE(14,4)
DO 1 I=1,C
EV(I)=SE(I,1)+SE(I,2)*T+SE(I,3)*T^2+SE(I,4)*T^3
CONTINUE
RETURN
END
SUBROUTINE DENV(T,EV)
INTEGER C,CT
COMMON C,CT,N
COMMON /A3/SE
DIMENSION EV(C),SE(14,4)
DO 1 I=1,C
EV(I)=SE(I,2)+SE(I,3)*2.*T+3.*SE(I,4)*T^2
CONTINUE
RETURN
END
SUBROUTINE DIST(T,AK)
INTEGER C,CT
COMMON C,CT,N
COMMON /A1/SA
DIMENSION AK(C),SA(14,4)
DO 1 I=1,C
AK(I)=T*(SA(I,1)+SA(I,2)*T+SA(I,3)*T^2+SA(I,4)*T^3)^3
CONTINUE
RETURN
END
SUBROUTINE DDIST(T,AK)
INTEGER C,CT
COMMON C,CT,N
COMMON /A1/SA
DIMENSION AK(C),SA(14,4)
DO 1 I=1,C
AK(I)=(SA(I,1)+SA(I,2)*T+SA(I,3)*T^2+SA(I,4)*T^3)^3+3.*T*(
1SA(I,2)+2.*SA(I,3)*T+3.*SA(I,4)*T^2)*(SA(I,1)+SA(I,2)*T
2+SA(I,3)*T^2+SA(I,4)*T^3)^2
CONTINUE
RETURN
END
```

CHE-1985-M-ISL-COM